

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky

## DIPLOMOVÁ PRÁCE

2020

Bc. Lukáš Krais

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

Optická kontrola průmyslového výrobku  
Optical Inspection of Industrial Product

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

## Zadání diplomové práce

Student: **Bc. Lukáš Krajs**  
Studijní program: N2649 Elektrotechnika  
Studijní obor: 2612T041 Řídicí a informační systémy  
Téma: **Optická kontrola průmyslového výrobku**  
**Optical Inspection of Industrial Product**  
Jazyk vypracování: čeština

### Zásady pro vypracování:

Při sériové výrobě průmyslových výrobků, tedy i elektronických součástek, je zapotřebí provádět kontrolu kvality hotového výrobku. Pro vizuální kontrolu se používá technologie nazvaná „strojové vidění“. Průmyslová kamera spolu s vhodným objektivem a osvětlením sejme obraz výrobku a přenese jej do počítače. V počítači je pak nasnímaný obraz z kamery zpracován pomocí SW pro strojové vidění. Tento SW dokáže nasnímaný obraz vyhodnotit dle požadavků zadavatele. Diplomová práce se zabývá výběrem vhodných komponent testeru a zejména vývojem SW komponent pro výše popsany typ testeru.

### Body zadání:

1. Rozbor problematiky zpracování obrazu v oblasti průmyslového testování.
2. Popis vývojových prostředí/technologií používaných pro návrh kamerových systémů - přehled a charakteristika.
3. Rozbor funkce a popis algoritmů strojového vidění pro požadované optické testy průmyslového výrobku.
4. Teoretický rozbor a návrh řešení kamerové inspekce průmyslového výrobku.
5. Návrh uživatelských rozhraní SW.
6. Návrh statistického zpracování výsledných dat.
7. Vývoj SW komponent v programovacím prostředí LabVIEW pro firemní testovací SW framework.
8. Nasazení a odladění vyvinutých komponent v průmyslové praxi.
9. Ověření funkce a zhodnocení.

### Seznam doporučené odborné literatury:

- [1] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. Praha: BEN - technická literatura, 2008, 247 s. ISBN 978-80-7300-245-9.
- [2] HOTAŘ, Vlastimil. *Úvod do problematiky strojového vidění*. V Liberci: Technická univerzita, 2015. ISBN 978-80-7494-156-6.
- [3] BRESS, Thomas J. *Effective labview programming*. 1st ed. Allendale: NTS Press, 2013, 701 s. ISBN 1934891088.
- [4] BITTER, Rick, Taqi MOHIUDDIN a Matt NAWROCKI. *LabView advanced programming techniques*. 2nd ed. Boca Raton: CRC Press, c2007, 499 s. ISBN 0849333253.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **doc. Ing. Petr Bilík, Ph.D.**

Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020



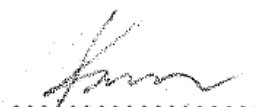
doc. Ing. Jiří Koziolek, Ph.D.  
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

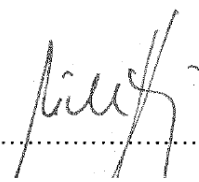
V Ostravě 5. května 2020

A handwritten signature in black ink, appearing to read 'Lukáš Krais', written over a horizontal dashed line.

Bc. Lukáš Krais

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 5. května 2020



.....

Ing. Michal Harhaj

## **Abstrakt**

Tato diplomová práce popisuje návrh a realizaci testeru pro optické kontroly v průmyslové výrobě. Diplomová práce se zabývá návrhem a realizací SW. Software testeru je tvořen v grafickém programovacím prostředí LabVIEW, při dodržení standardů pro integraci firemního testovacího SW frameworku. Výsledný software či jeho části jsou po otestování nasazeny v průmyslové výrobě.

**Klíčová slova:** strojové vidění, LabVIEW, zpracování obrazu, optická kontrola, zásuvný modul, zařízení

## **Abstract**

This diploma thesis describes design and realization of tester for optical inspection in industrial production. This thesis deals with the design and implementation of software. The tester software is created in the LabVIEW graphical programming environment, while adhering to standards for integration of the company's testing SW framework. The complete software or its parts are after testing deployed in industrial production.

**Key Words:** machine vision, LabVIEW, image processing, optical inspection, plugin, device

## Obsah

Seznam použitých symbolů a zkratk .....	10
Seznam ilustrací .....	11
Seznam rovnic .....	13
Úvod.....	14
1 Rozbor problematiky zpracování obrazu v oblasti průmyslového testování.....	15
1.1 Strojové vidění .....	15
1.2 Kamery .....	15
1.2.1 Rozdělení kamer podle způsobu snímání.....	16
1.2.2 Rozdělení kamer podle technologie snímače .....	17
1.2.3 Rozdělení kamer podle komunikačního rozhraní.....	18
1.3 Objektivy.....	19
1.4 Osvětlovače .....	22
1.5 Filtry.....	22
2 Popis vývojových prostředí/technologií používaných pro návrh kamerových systémů – přehled a charakteristika .....	24
2.1 LabVIEW .....	24
2.2 Vision Builder for Automated Inspection .....	25
2.3 VisualApplets.....	26
3 Rozbor funkce a popis algoritmů strojového vidění pro požadované optické testy průmyslového výrobku .....	28
3.1 ROI – Region of interest .....	28
3.2 Hranový detektor.....	28
3.2.1 Gaussův filtr .....	29
3.2.2 První derivace Gaussovy funkce .....	29
3.2.3 Nalezení lokálních maxim.....	29
3.2.4 Eliminace nevýznamných hran .....	29
4 Popis používaných sběrnic při praktickém návrhu SW ovladače pro zařízení HSX.....	31
4.1 HSX.....	31
4.2 Sběrnice CAN .....	31
4.2.1 Typy zpráv CAN protokolu.....	32
4.2.2 Popis datového rámce zpráv.....	32
4.3 Sběrnice LIN .....	32



4.3.1	Popis datového rámce zprávy .....	33
4.3.2	LIN řídicí zařízení (Master) .....	33
4.3.3	LIN podřízené zařízení (Slave) .....	33
5	Teoretický rozbor a návrh řešení kamerové inspekce průmyslového výrobku .....	35
5.1	Návrhu řešení kamerové inspekce pro kontrolu skla automobilu .....	35
5.1.1	Použité komponenty .....	35
5.1.2	Studie proveditelnosti .....	35
5.1.3	Kamerová sestava, nainstalována v průmyslové výrobě .....	36
6	Návrh uživatelských rozhraní SW .....	38
6.1	Popis rozhraní SW Frameworku pro testování .....	38
6.2	Postup při návrhu GUI .....	39
6.3	Rozložení prvků panelu a interpretace popisků .....	40
7	Návrh statistického zpracování výsledných dat .....	42
7.1	Naměřená data mohou být uložena do souboru s příponou .txt .....	42
7.2	Naměřená data mohou být uložena do souboru s příponou .pdf .....	42
7.3	Report měření nanesené tloušťky krycí vrstvy laku na krycím skle světlometu .....	43
8	Vývoj SW komponent v programovacím prostředí LabVIEW pro firemní testovací SW Framework .....	45
8.1	SW modul kontroly nanesené pasty na skle automobilu .....	45
8.1.1	GUI SW modulu pro kontrolu nanesené pasty .....	45
8.1.2	Implementace kódu SW modulu .....	46
8.1.3	Nastavení parametrů SW modulu .....	47
8.2	SW ovladač pro komunikaci se zařízením HSX .....	47
8.2.1	GUI SW ovladače .....	48
8.2.2	Implementace kódu SW ovladače .....	48
8.2.3	Nastavení parametrů SW ovladače .....	49
8.2.4	Testovací aplikace pro HSX .....	49
8.3	Tvorba SW ovladačů pro zařízení kontroly světelné stopy .....	50
8.3.1	SW ovladač dálkoměru .....	51
8.3.2	SW ovladač ionizátoru .....	53
8.4	Algoritmus pro vykreslování křivek, textu .....	56
9	Nasazení a odladění vyvinutých komponent v průmyslové praxi .....	59
9.1	Parametrizace PC, instalace SW .....	59
9.2	Testování .....	59

10	Ověření funkce vyvinutého SW a zhodnocení .....	61
	Závěr .....	62
	Literatura .....	63
	Přílohy .....	65

## **Seznam použitých symbolů a zkratek**

CAN – Control Area Network (řídící sběrnice)

LIN – Local Interconnect Network (podřízená, místní sběrnice)

SW – Software

GUI – Graphical User Interface (grafické uživatelské rozhraní)

DTS – Diagnostics Tool Set (sada diagnostických nástrojů)

Device – SW ovladač zařízení komunikující s PC v aplikaci

Plugin – SW modul aplikace pro manipulaci s daty

PoE – Power Over Ethernet (napájení po ethernetu)

ROI – Region Of Interest (oblast zájmu)

VI – Virtual Instrument (virtuální přístroj)

LED Modul – dílčí část světlometu

Interface – rozhraní zajišťující správný přenos dat mezi zařízeními

PDF – Portable Document Format (přenosný formát dokumentu)

SQL – Structured query language (strukturovaný dotazovací jazyk)

Overlay – překrytí obrázku, textu, bitmap bez zničení obrazových dat

CCD – Charge-Couple-Device (zařízení s vázanými náboji)

CMOS – Complementary Metal–Oxide–Semiconductor (komplementární polovodič oxidu kovu)

USB – Universal Serial Bus (univerzální sériová sběrnice)

BNC – Bayonet Neill Concelman (vysokofrekvenční konektor koaxiálních kabelů)

MTF – Modulation Transfer Function (funkce modulace přenosu)

HSX – diagnostické rozhraní řady HS

## Seznam ilustrací

Obrázek 1 Kamerový a datový řetězec .....	16
Obrázek 2 Řádková kamera .....	16
Obrázek 3 Maticová kamera .....	17
Obrázek 4 CCD vs. CMOS snímače [7] .....	17
Obrázek 5 GigE kamera [17] .....	18
Obrázek 6 USB 3.0 kamera [17] .....	18
Obrázek 7 Camera Link kamera [17] .....	19
Obrázek 8 CoaXPress kamera [14] .....	19
Obrázek 9 Ohnisková vzdálenost .....	20
Obrázek 10 Rozdílná hloubka ostroty s upravenou clonkou (nalevo – otevřená clona, napravo – téměř uzavřená clona) .....	20
Obrázek 11 Zapnutá korekce vinětace obrazu (vlevo), vpravo vypnutá [16] .....	21
Obrázek 12 Funkce modulace přenosu [15] .....	22
Obrázek 13 Zadní osvětlení, difúzní osvětlení, směrové osvětlení, osvětlení typu temné pole [8][4] ..	22
Obrázek 14 Použití modrého filtru při chromatické vadě objektivu (Chromatická aberace) [19] .....	23
Obrázek 15 Knihovna pro zpracování souborů .....	24
Obrázek 16 Čelní panel aplikace ve vývojovém prostředí LabVIEW .....	25
Obrázek 17 Příklad aplikace v prostředí Vision Builder AI .....	26
Obrázek 18 Příklad kódu v technologii VisualApplets [20] .....	27
Obrázek 19 Část kódu a vizualizace hledání oblastí pomocí volných tvarů [9] .....	28
Obrázek 20 Připojení jednotky na sběrnici CAN [5] .....	31
Obrázek 21 Datový rámec CAN sběrnice [5] .....	32
Obrázek 22 Připojení jednotky na sběrnici LIN [5] .....	33
Obrázek 23 Datový rámec sběrnice LIN [6] .....	33
Obrázek 24 Základní datové pole zprávy [6] .....	33
Obrázek 25 Řízení přístupu zařízení na sběrnici LIN [6] .....	34
Obrázek 26 Umístění komponent kamerové sestavy – temné pole [4] .....	36
Obrázek 27 Snímek skla při studii .....	36
Obrázek 28 Sestava inspekce pro detekci teplo-vodivé pasty na obvodu skla .....	37
Obrázek 29 GUI pro nový SW ovladač .....	39
Obrázek 30 Srovnání dvou odlišných SW ovladačů se stejným GUI .....	40
Obrázek 31 Část obsahující nastavení pozice prvků čelního panelu podle zvoleného rozlišení .....	41
Obrázek 32 Knihovny pro ukládání souborů *.pdf [13] .....	43
Obrázek 33 Nastavení obsahu a vzhledu textu pomocí knihoven Exaprom PDF [13] .....	43
Obrázek 34 Report měření tloušťky laku s nastavenými parametry .....	44
Obrázek 35 Čelní panel SW modulu kontroly nanášené pasty .....	46
Obrázek 36 Blokový diagram kroků daného algoritmu .....	47
Obrázek 37 SW ovladač zařízení pro komunikaci dle ŘJ světlometu .....	48
Obrázek 38 Algoritmus čtení příkazů v blokovém diagramu testovací aplikace HSX .....	50
Obrázek 39 Testovací panel pro komunikaci s ŘJ světlometu .....	50
Obrázek 40 Stanice pro kontrolu světelné stopy .....	51
Obrázek 41 Část kódu pro zpracování přijímaných dat ve formě 3bytového pole .....	52
Obrázek 42 GUI panel SW ovladače .....	52
Obrázek 43 Princip komunikace s dálkoměrem .....	53
Obrázek 44 UML diagram popsaného zařízení .....	54

Obrázek 45 GUI SW ovladače ionizátoru.....	55
Obrázek 46 Logické zpracování řízného čištění .....	56
Obrázek 47 Panel pro sestavení sekvence optických testů.....	57
Obrázek 48 Vykreslování os do obrazu s převodem barevných hodnot pixelů .....	58
Obrázek 49 Vykreslování jednotek na osy.....	58

## Seznam rovnic

1 Gaussova filtrační funkce.....	29
2 Gaussova první derivace .....	29
3 Segmentace prahování.....	30
4 Globální prahování.....	30
5 Prahování s hysterezí.....	30

## Úvod

Průmyslový proces výroby je dnes z pohledu kvality, ceny a rychlosti vyrobených kusů přesně definován. Výrobky jsou nejčastěji vyráběny sériově, přičemž je kladen důraz na snížení nákladů za předpokladu precizní výroby. Díky tomu se vizuální inspekce stala nedílnou součástí výrobního procesu. Vizuální kontrola výrobku se postupem času posouvá do oblasti automatizované kontroly výrobku. Pomocí kamery je možné ji nově aplikovat ve fázi procesu, kde lidská kontrola nebyla možná. Oblast strojové inspekce či strojového vidění umožňuje zrychlení kontroly výrobku a odhalení nedostatků již v průběhu výroby.

Kamerový a datový řetězec se skládá z průmyslové kamery s použitím vhodného osvětlovače a nadřazeným systémem pro zpracování dat PC. Kamera sejme obraz, který je zaslán prostřednictvím komunikačního rozhraní do PC, kde je následně zpracován, analyzován a vyhodnocen. Analýza a vyhodnocení obrazu musí být provedeno co nejrychleji a co nejpresněji.

Teoretická část diplomové práce se zabývá rozбором kamerového řetězce, popisem hardware, přehledem vývojových prostředí pro zpracování obrazu, matematickou interpretací hranového detektoru, který je uplatněn v praktické části při vývoji SW komponenty SW modulu kontroly nanesené pasty na skle automobilu. Popisem principu komunikačních sběrnic CAN a LIN používaných v praktické části při řešení vývoje SW ovladače pro komunikaci řídicí jednotky světlometu automobilu a průmyslového PC (SW Framework).

Praktická část diplomové práce začíná kapitolou s názvem Teoretický rozbor a návrh řešení kamerové inspekce průmyslového výrobku. Tato část je věnována konkrétnímu výběru HW komponent pro kontrolu průmyslového výrobku, návrhu řešení vývoje uživatelských rozhraní pro SW Framework, návrhu statistického zpracování dat a vývoji SW komponent využitých při optické kontrole průmyslového výrobku v průmyslové výrobě.

# 1 Rozbor problematiky zpracování obrazu v oblasti průmyslového testování

## 1.1 Strojové vidění

Strojové vidění se většinou používá v průmyslové oblasti při zpracování obrazové informace z kamery pomocí počítače. Algoritmus zpracování obrazu je poté aplikován na každý kontrolovaný objekt ve formě vstupní informace pro rozhodování informačního systému. Nejčastější aplikací strojového vidění je kamerová kontrola při kompletaci výrobku. Na základě kamerového testu a použití algoritmu dojde k označení či třídění neúplných výrobků a vadných kusů. [3]

Úlohy strojového vidění:

- Čtení QR, 2D a 3D kódu
- Třídění a kategorizace výrobků
- Kontrola potisku
- Měření povrchu 3D výrobků
- Detekce vad povrchu výrobku

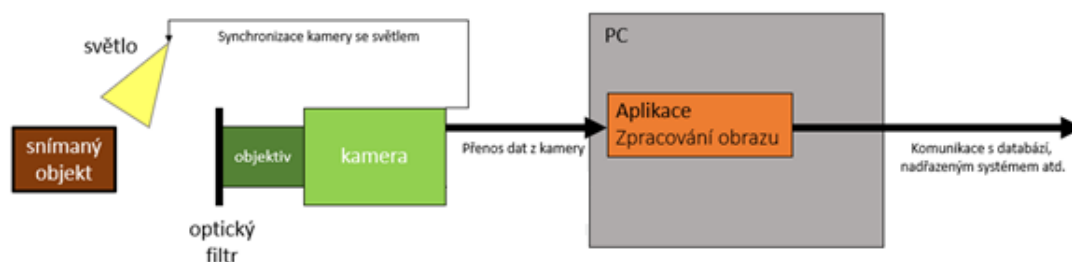
## 1.2 Kamery

Kamera zachytí obraz snímaného předmětu, systém analýzy obrazu jej vyhodnotí podle přesně definovaného algoritmu a provede požadovanou akci na základě výsledku vyhodnocení. V dnešní době je jediným efektivním a prakticky možným způsobem vyhodnocení obrazové informace počítač. Pro strojové vidění je typické, že se výměna získaných informací s okolím provádí na základě zpětné vazby s výrobním procesem. Na jedné straně se získají vstupní informace a na druhé straně se na základě výsledků analýzy obrazu provádí různé úkony, kterými může být například vyřazení nevhodného či špatně vyrobeného výrobku, přesnost při plnění lahví, kontrola přítomnosti etiket a další. Informace o splnění požadovaného úkonu jde pomocí digitálních vstupů přímo do zařízení nebo je informace předána řídicímu systému, který již sám provede potřebný zásah. [1]

Nejdůležitějším požadavkem na kameru je spolehlivost, odolnost v průmyslovém prostředí a nepřetržitý provoz. V extrémních případech je vyžadována vodotěsnost a odolnost vlivu vysokých teplot. Proto většina průmyslových kamer splňuje takové požadavky v základním provedení.

Nejčastější kamerový řetězec je složen z následujících částí. Světelný zdroj, který osvětluje snímaný objekt (testovaný výrobek). Obraz je vykreslován pomocí optické soustavy na světlo citlivý senzor kamery, která obraz převádí do digitální podoby, kde je poslán přes komunikaci do nadřazeného systému (PC) pro zpracování a vyhodnocení (Obrázek 1). Posledním krokem je úkon, který má být proveden: uložení do databáze, zpráva operátorovi, zpráva řídicímu systému. [3]





Obrázek 1 Kamerový a datový řetězec

Kamery lze rozdělit dle několika vybraných kritérií:

- Podle způsobu snímání
- Podle technologie snímače
- Podle komunikačního rozhraní

### 1.2.1 Rozdělení kamer podle způsobu snímání

Maticová a řádková kamera se výrazně liší ve způsobu použití a snímání obrazu.

#### Řádková kamera

Řádkovou kameru použijeme pro snímání nekonečných pásů či rotujících předmětů. Pracuje na stejném principu jako skener, snímá řádek po řádku vysokou frekvencí až 140kHz. Celá plocha pixelů je vyhrazena pro světlo-citlivou buňku a elektronika pro zpracování je umístěna bokem. Její citlivost je z tohoto důvodu 2x vyšší oproti citlivosti klasické kamery (Obrázek 2).



Obrázek 2 Řádková kamera

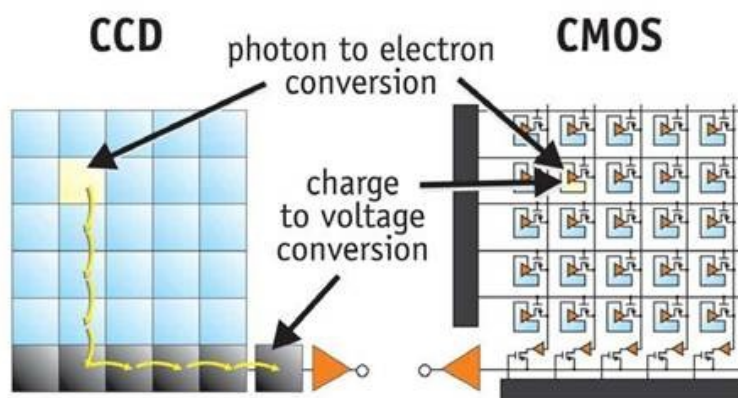
## Maticová kamera

Maticovou kameru použijeme všude tam, kde potřebujeme najednou sejmout obraz jednotlivého předmětu. Snímá obraz v dimenzích x a y (Obrázek 3).



Obrázek 3 Maticová kamera

### 1.2.2 Rozdělení kamer podle technologie snímače



Obrázek 4 CCD vs. CMOS snímače [7]

#### CCD

Obsahuje světlo-citlivé buňky, které při hromadění elektronů vytváří elektrický náboj úměrný počtu fotonů dopadajících na světlo-citlivé buňky. Data jsou čtena po řádcích. CCD snímač obsahuje posuvný registr, do kterého se postupně řádky přesouvají, poté prochází přes zesilovač do A/D převodníku, ze kterého vyjdou digitální data. [7]

Výhody:

- Nízký šum
- Kvalita obrazu

## CMOS

U tohoto typu snímače jsou data přenášena z každého pixelu samostatně, tedy každý pixel je zpracován zvlášť konkrétním elektrickým obvodem. Proto je označován zkratkou APS (Active-Pixel Sensor). Tento typ snímače je v dnešní době nejpoužívanější, a to díky nízké hodnotě šumu při vysoké citlivosti a většímu dynamickému rozsahu. [7]

Výhody:

- Nízká spotřeba
- Rychlost snímání a zpracování dat

### 1.2.3 Rozdělení kamer podle komunikačního rozhraní

#### GigE

Gigabitový Ethernet lze označit za nejpoužívanější komunikační rozhraní v aplikacích strojového vidění. A to díky robustní kabeláži, nízkému rušení a jednoduchému připojení do síťové infrastruktury. Napájení samotné kamery je řešeno rozhraním PoE (Obrázek 5). Díky tomu odpadá další kabeláž ke kameře, komunikace i napájení je řešeno po ethernetovém kabelu. Hlavní použití tohoto typu rozhraní je v aplikacích strojového vidění, kdy je zapojeno několik kamer v jednom zařízení a každá plní určitou funkci pro následné zpracování obrazu a vyhodnocení testu. [17]



Obrázek 5 GigE kamera [17]

#### USB 3.0

USB 3.0 je rozhraní s velmi snadným připojením a vysokou propustností. Komunikace USB3.0 (Obrázek 6) zajišťuje vysokou spolehlivost přenosu dat a nízkou zátěž CPU. Nejčastějším použitím kamery s tímto rozhraním je vysokorychlostní snímání ve vestavěných systémech. [17]



Obrázek 6 USB 3.0 kamera [17]

## Camera Link

Robustní rozhraní komunikace Camera Link je určeno pro aplikace s požadavky nízké latence a vysokoobjemovým přenosem dat (řádkové kamery). Pro použití Camera Link připojení (Obrázek 7) je potřeba mít v počítači nainstalovanou kartu s daným rozhraním (tzv. frame grabber). Důvodem je přístup k nastavení parametrů kamery v jiném programu. [18]



Obrázek 7 Camera Link kamera [17]

## CoaXPress

Rozhraní CoaXPress kamery (Obrázek 8) je kombinací klasického koaxiálního kabelu a moderních vysokorychlostních technologií pro sériový přenos dat. CoaXPress je způsob profesionálního použití v průmyslových aplikacích, lékařství či biologii. Deklarovaná rychlost, kterou může tento standard dosahovat je 12,5Gbit/s do vzdálenosti 30 m. [21]



Obrázek 8 CoaXPress kamera [14]

## 1.3 Objektivy

Objektiv je soustava čoček a slouží pro soustředění světla na světlo-citlivý senzor. Podle typu závitu lze rozdělit objektivy na C-mount (pevné ohnisko), CS-mount (zoom) a F-mount (velkoformátový). Objektivy pro práci v průmyslovém prostředí musí být vysoce odolné vůči nečistotám a vibracím. Jsou vybaveny mechanickým zámekem pro nastavení parametrů jako jsou hloubka ostrosti či „clona“ (množství dopadajícího světla na senzor). Většinou jsou objektivy vybírány podle vzdálenosti objektu od kamery a použitého typu snímače.

### Výběr objektivu dle typu aplikace:

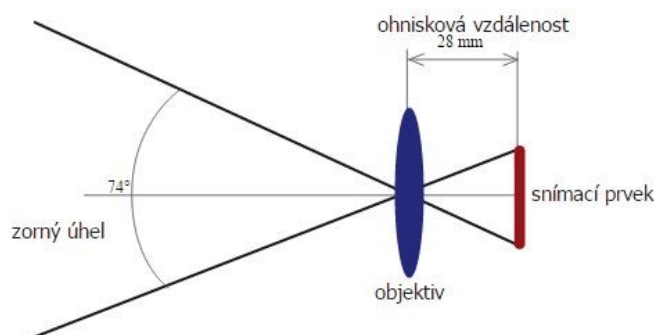
- Vzdálené předměty – teleobjektiv

- Miniaturní předměty (SMD součástka) – Makro objektiv
- Snímání velké části prostoru – širokoúhlý objektiv, v extrémním případě tzv. „Rybí oko“

## Vybrané parametry objektivů

### Ohnisková vzdálenost

Ohnisková vzdálenost je obrazová vzdálenost středu čočky od ohniska. Místo, kam dopadají paprsky na světlo-citlivý senzor. Značí se písmenem  $f$  – ohnisková vzdálenost (Obrázek 9) určuje zorné pole a zvětšení snímaného objektu.



Obrázek 9 Ohnisková vzdálenost

### Hloubka ostrosti

Vzdálenost okolo bodu, kdy se pozorovateli objekt v této vzdálenosti jeví jako ostrý. Hloubka ostrosti je proměnný faktor (Obrázek 10) a lze ji přizpůsobit dle potřeb uživatele. Hloubku ostrosti lze upravit tak, že předměty v pozadí budou neostře a předměty v popředí, které nás zajímají si zaostříme, či naopak. [2]



Obrázek 10 Rozdílná hloubka ostrosti s upravenou clonkou (nalevo – otevřená clona, napravo – téměř uzavřená clona)

## **Clona**

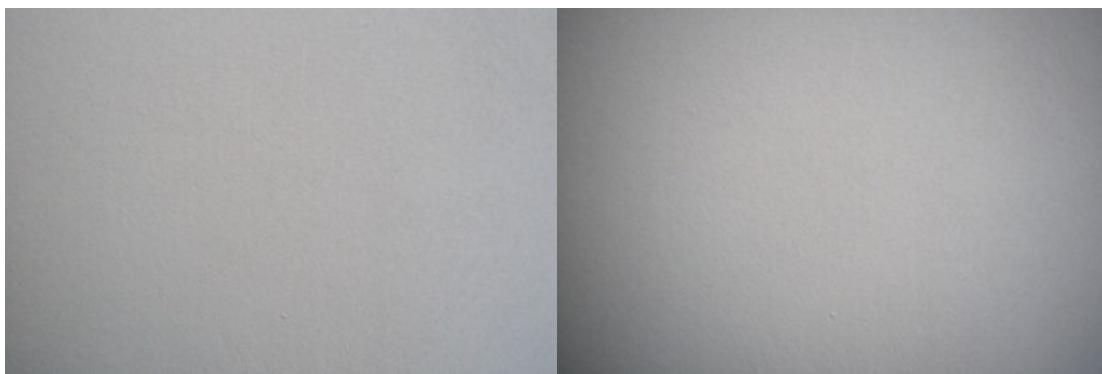
Clona určuje množství dopadajícího světla na čip. Ovlivňuje hloubku ostroty. Pokud clonu úplně uzavřeme nebude propouštět objektiv žádné dopadající světlo a obraz tak bude absolutně tmavý.

## **Světelnost**

Světelnost objektivu je parametr, který udává množství propuštěného světla na čip. Hodnota světelnosti je udávána clonovým číslem, vyjadřující poměr ohniskové vzdálenosti k průměru vstupní pupily. Pro objektiv se vždy uvádí nejvyšší možná hodnota světelnosti. [17]

## **Vinětace objektivu**

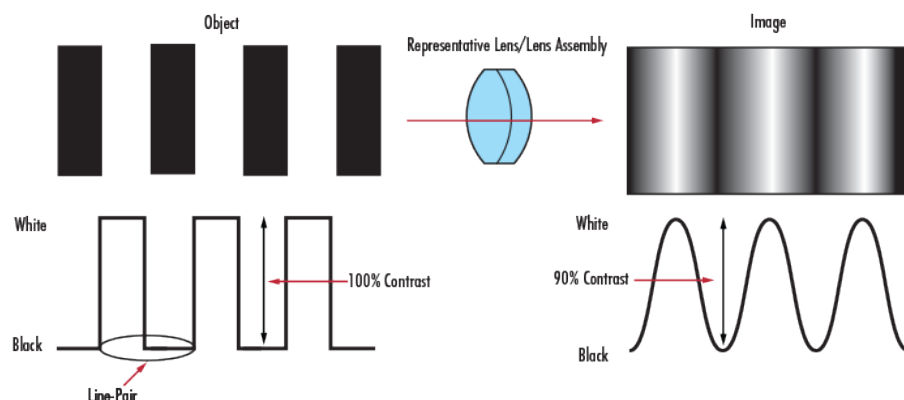
Vinětace objektivu je vada, při které dochází ke změně (snížení) jasu v oblasti krajních pixelů obrazu. Obraz se poté jeví jako zakulacený, simulující vnořený filtr do optické soustavy. Vinětaci je možno zmírnit vyšší clonou objektivu, ale za předpokladu této možnosti. Zlepšit samotný obraz lze pomocí úpravy již pořízeného obrazu nebo zvýšení saturace obrazu či použitím polarizačního filtru. [16]



Obrázek 11 Zapnutá korekce vinětace obrazu (vlevo), vpravo vypnutá [16]

## **Funkce modulace přenosu (MTF)**

Funkce modulace přenosu je míra schopnosti optického systému přenášet různé úrovně detailů z objektů v obrazu. Výkon je měřen jako kontrast (úroveň šedé) nebo modulace, produkovaný pro dokonalý zdroj této úrovně detailu. Při měření MTF se posuzuje míra degradace kvality snímku testovacího obrazce. Posuzuje se tedy výstup z objektivu oproti snímané předloze. MTF je zároveň způsob, jak sloučit rozlišení a kontrast do jedné specifikace. [15]



Obrázek 12 Funkce modulace přenosu [15]

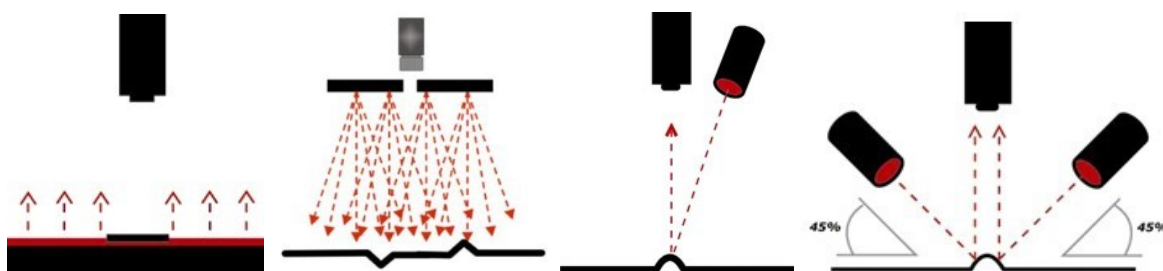
## 1.4 Osvětlovače

Osvětlovač je nedílnou součástí kamerového řetězce. Uplatňuje se ve většině aplikací strojového vidění pro zvýraznění hledaných artefaktů. Pomocí správného nasvícení objektu lze zvýšit vliv na kvalitu obrazové informace. Snímaný obraz nemusí být dokonalý, je však potřeba správně nasvítit hledané prvky pro zesílení jejich odlišností (maximalizaci kontrastu) vůči ostatním částem obrazu. Při výběru osvětlovače je zapotřebí zvážit několik kritérií: tvar a povrch objektu, umístění snímaného objektu a barvu objektu. Zvolit správný osvětlovač je mnohdy složitější než výběr kamery a objektivu, u kterých jsou parametry výběru přesně definovány. [4]

Světelné zdroje lze rozdělit na dva typy:

- Monochromatické – světlo o jedné vlnové délce (laser, LED)
- Polychromatické – světlo složené z množství monochromatických světél, např. bílé světlo (žárovka, bílé LED)

Typy osvětlovačů dle umístění (Obrázek 13):



Obrázek 13 Zadní osvětlení, difúzní osvětlení, směrové osvětlení, osvětlení typu temné pole [8][4]

## 1.5 Filtry

Filtry jsou zařízení sloužící pro potlačení určité vlnové délky světla. Filtry se ve většině aplikací používají pro zvýraznění kontrastu černobílých snímků kamery. Při zpracování barevného snímku se využívají pro oddělení UV a NIR části spektra. V některých případech slouží jako ochrana objektivu.



## Barevné filtry

Barevné filtry jsou filtry, které jsou barevně odlišitelné lidským okem. Modrá a zelená barva filtru zvyšuje přesnost měření a kontrast monochromatického snímku zachyceného kamerou. Propouští kratší vlnové délky (Obrázek 14). Červená a oranžová barva filtru propouští delší vlnové délky, a proto se tyto typy filtru ve strojovém vidění často nepoužívají.



Obrázek 14 Použit modrého filtru při chromatické vadě objektivu (Chromatická aberace) [19]

## Pásmové filtry

Propouštějí vlnové délky definovaných hodnot, ostatní vlnové délky blokují. Pásmové filtry obsahují velmi tenkou vrstvu, kterou je filtr pokryt a určuje jeho optické vlastnosti. Vlnové délky barevných filtrů jsou v rozsahu 465nm až 865nm.

## Polarizační filtry

Většina světelných zdrojů vyzařuje nepolarizované světlo. V aplikaci strojového vidění je potřeba eliminovat odrazivost světelného paprsku od povrchu objektu. Polarizační vrstva se nachází mezi dvěma skly filtru, jedná se o polymerovou fólii, která je citlivá na vlhkost.[19]



## 2 Popis vývojových prostředí/technologií používaných pro návrh kamerových systémů – přehled a charakteristika

Zpracování obrazu lze řešit v řadě vývojových prostředích, která jsou přizpůsobena k tomuto účelu (knihovny, ovladače, nástroje). Při hledání vhodného prostředí je zapotřebí myslet nejen na jednoduchý návrh zpracování dat z kamery, ale také na obsažené knihovní funkce pro zpracování obrazu.

Při výběru vývojového prostředí pro návrh kamerové inspekce je zapotřebí zohlednit požadavky zákazníka a potřeby vizualizace této inspekce. Pokud se jedná o jednoduchou aplikaci, lze použít intuitivní prostředí Vision Builder, který umožňuje rychlou orientaci v paletě funkcí, pro návrh kamerové inspekce a vizualizaci snímaných částí obrazu.

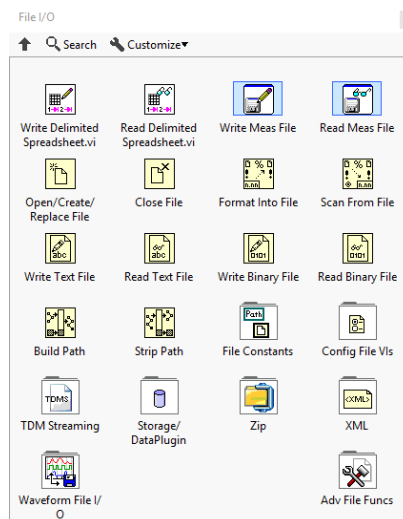
Pokud je známo, že po návrhu systému neproběhnou žádné změny a nebude potřeba tak měnit parametry zadání, lze systém pro zpracování obrazu napsat zcela přesně na daný projekt. Můžou zde však být aplikace, u kterých je požadováno, aby byla možná další integrace. Popřípadě je možno systém navrhnout tak, aby při změně typu výrobku, avšak při zachování stejného principu měření, nedocházelo k velké změně aplikace. To znamená, že například při testování svítivosti jednotlivých LED ve světlometru nebude problém testovat ve stejné aplikaci jiný druh světlometru po úpravě určitých parametrů. [21]

### 2.1 LabVIEW

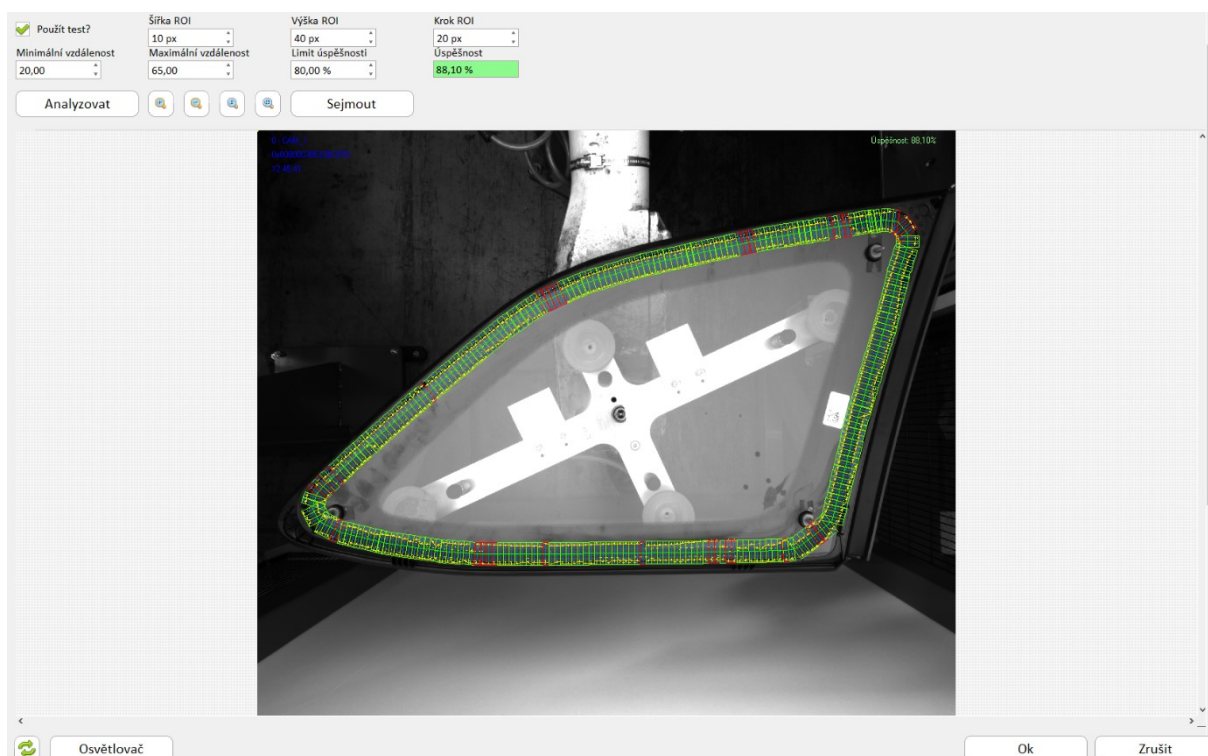
Vývojové prostředí LabVIEW od společnosti National Instruments využívá principu přenosu dat označující termínem data-flow, což znamená, že vstupní data vstupují do systému zleva a výstupní data pro zpracování vychází ze systému směrem doprava. Knihovní funkce jsou potřebné pro ovládání připojeného hardwaru, zpracování naměřených dat a jejich vizualizaci (Obrázek 16) či přípravu pro další zpracování. Uspadňuje integraci hardwaru od různých výrobců, reprezentaci komplexní logiky na diagramy, vývoj analýzy zpracování dat a návrh uživatelských rozhraní pro technické inženýrství. Výhodou tohoto programovacího prostředí jsou knihovny. Většina z nich je již obsažena ve vývojovém prostředí a další jsou ke stažení či k zakoupení. Tyto knihovny (Obrázek 15) jsou používány pro velmi rychlé sestavení aplikace. Knihovny obsahují již vytvořené části kódu pro specifické funkce.[1], [3]

Například uložení dat do textového souboru:

- Otevření / vytvoření souboru
- Uložení dat v požadovaném formátu
- Kontrola zapsaných dat
- Zavření souboru



Obrázek 15 Knihovna pro zpracování souborů



Obrázek 16 Čelní panel aplikace ve vývojovém prostředí LabVIEW

## 2.2 Vision Builder for Automated Inspection

Vision Builder je aplikační software sloužící pro rychlý vývoj a nasazení systému kontroly strojového vidění. Pomocí vývojových nástrojů lze rychle konfigurovat kamery, vyvinout algoritmus testování a inspekční kroky, a generovat tak výsledky inspekce (Obrázek 17). Nejedná se o čisté programování, ale o postupné skládání již naprogramovaných bloků ve formě funkcí. Funkce jsou rozděleny do logických záložek, ve kterých se uživatel rychle zorientuje.

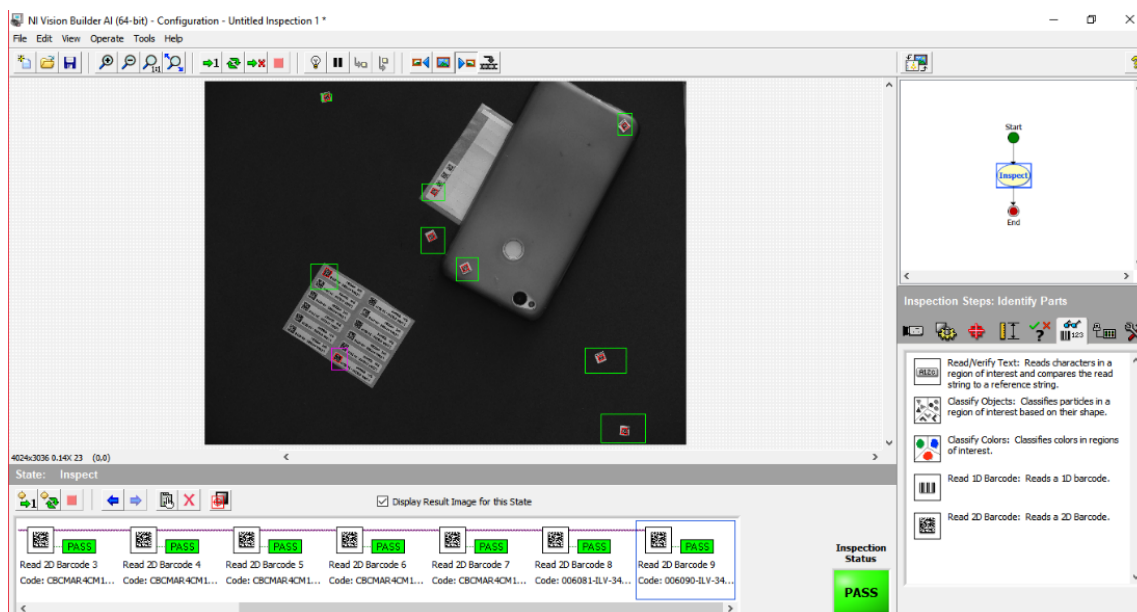
Vizuální inspekce v tomto prostředí lze rozdělit do několika bloků:

- Sejmout snímek z kamery
- Upravit snímek
- Zpracování obrazu dle požadavků (měření jasu, detekce hran, hledání objektů, měření vzdáleností, čtení kódů – barcode, QR)
- Komunikace s okolím DI, DO, sériová linka, Ethernet
- Uložit snímek
- Propojení kódu s LabVIEW

Základní funkce:

- Hledání a porovnávání vzorů
- Čtení a ověřování písmen a znaků
- Kontrola barev (odstín, RGB v pixelu)
- Obrazová analýza (filtry)
- Klasifikace objektů v obraze

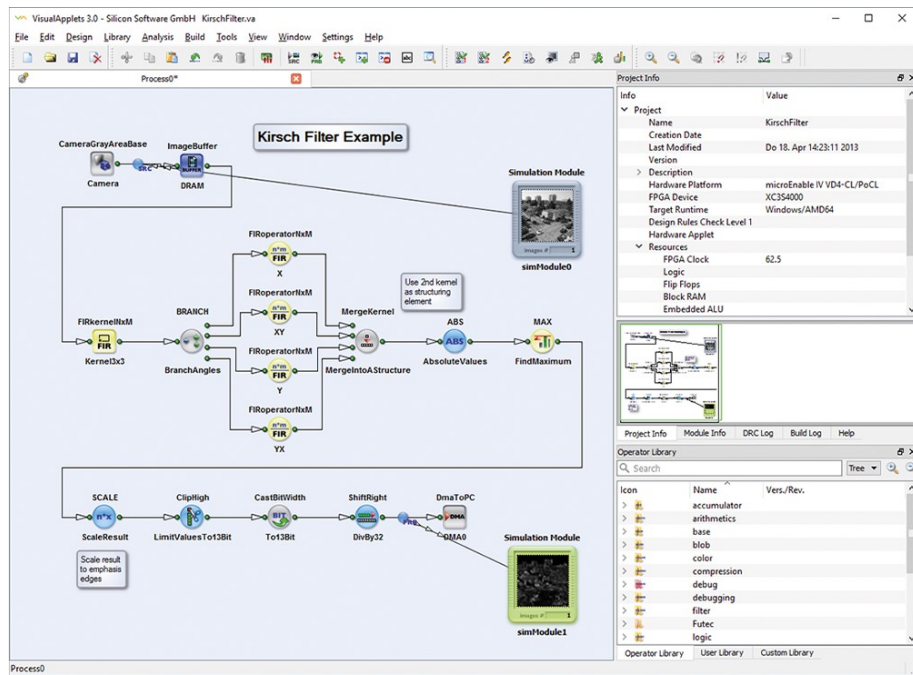
- Detekce a kontrola hran
- Měření geometrických vlastností (rozměry, vzdálenosti, úhly, plochy)
- Čtení čárových kódů (QR, 2D a 3D, laserové kódy)
- Logické a aritmetické funkce
- Kalibrace



Obrázek 17 Příklad aplikace v prostředí Vision Builder AI

## 2.3 VisualApplets

VisualApplets je integrované vývojové prostředí společnosti Basler pro zpracování obrazu běžící v reálném čase na FPGA (hradlové pole). Toto řešení se používá v mnoha průmyslových aplikacích, kde je potřeba zpracovat velké množství dat v reálném čase. VisualApplets umožňují uživatelský přístup k FPGA procesorům na různém hardwaru, jako jsou tzv. frame grabbery a průmyslové kamery. Stejně jako LabVIEW i VisualApplets používá grafické programování na principu data-flow (Obrázek 18). U průmyslových aplikací se většinou jedná o spolupráci této technologie a prostředí LabVIEW, kdy jsou snímky z kamery zpracovávány na FPGA a samotná aplikace informuje operátora o výsledku měření, která je již implementována v prostředí LabVIEW. [20]



Obrázek 18 Příklad kódu v technologii VisualApplets [20]

### 3 Rozbor funkce a popis algoritmů strojového vidění pro požadované optické testy průmyslového výrobku

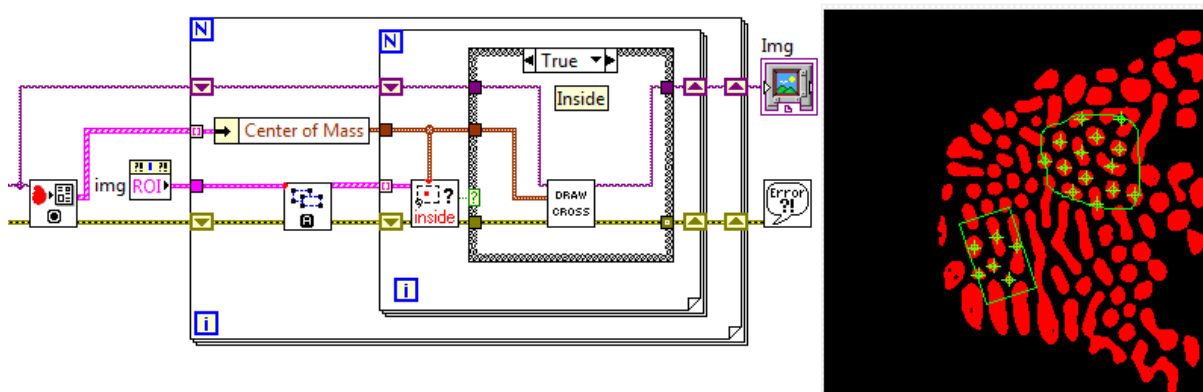
Následující kapitola je věnována teoretickému popisu algoritmů a funkcí použitých v praktické části diplomové práce při vytváření SW modulu kontroly nanesené pasty skla automobilu. Optické testy jsou využívány při kontrole částí obrazu. Části, které jsou pro detekci zajímavé označujeme jako oblasti zájmu. Definováním oblastí dojde k urychlení procesu vyhodnocení algoritmu pro daný optický test. Při vývoji SW modulu detekující teplo-vodivou pastu na skle automobilu bylo využito matematické funkce hledání hran v obraze. Funkce hranového detektoru byla přizpůsobena a nasazena v optickém testování výrobku při výstupní kontrole.

#### 3.1 ROI – Region of interest

Oblast zájmu je část obrazu, která je pro vyhodnocení či zpracování obrazu nejdůležitější. Umožňuje nám soustředit veškeré výkonné prostředky do části obrazu a zpracovat tak vyhodnocení v kratším čase. Vyhodnocení celého obrazu by znamenalo aplikovat výpočetní algoritmy na násobně větší množství obrazových dat. Vytvoření oblasti zájmu znamená zmenšení velikosti obrazu a eliminaci pixelů, které jsou pro analýzu nepodstatné.

Definovat tuto oblast ve vývojovém prostředí LabVIEW je v podstatě jednoduché. Nástroj pro vytvoření ROI uživatelem počítá s různými požadavky, a proto umožňuje vytvářet různé geometrické útvary. Většinou se oblast zájmu vytváří na začátku řetězce algoritmu zpracování obrazu, následně se pracuje s již ořezanou oblastí.[4]

Oblast zájmu lze nastavit i v samotném snímání obrazu kamerou bez jakýchkoli zpracování. Například některé IP kamery umožňují definovat tuto oblast pro detekci obličejů, registračních značek apod. Umožňují tak detekovat například jedoucí vozidla v koloně nebo počítat jejich průjezdnost či podávat informace o dopravní situaci. [9], [2]



Obrázek 19 Část kódu a vizualizace hledání oblastí pomocí volných tvarů [9]

#### 3.2 Hranový detektor

Algoritmus hranového detektoru obsahuje několik kroků pro získání nejlepšího výsledku hledání hran ve dvourozměrném diskretním obraze. Hranový detektor byl navržen tak, aby splňoval všechny hlavní kritéria: minimální chybovost, lokalizace a jednoznačná odezva. Detekce hran se využívá pro nalezení oblastí snímku, kde dochází k velkým změnám v intenzitě sousedních bodů. Na základě těchto změn lze nalézt kontury objektu a poté měřit jeho rozměry. Hrana je vlastnost přisouzená jednomu bodu v obraze

a vypočítá se z obrazové funkce v jeho okolí. Obrazová funkce je vektorová veličina, se svou velikostí a směrem.[9]

Samotná realizace vybraného Cannyho detektoru je realizována v jednotlivých krocích:

- Odstranění šumu
- Určení gradientu
- Nalezení lokálních maxim
- Eliminace nevýznamných hran

### 3.2.1 Gaussův filtr

Metoda se využívá při vyhlazení obrazu, odstranění detailů a šumu. Situace, při které máme k dispozici pouze jeden snímek a nelze tak použít průměrování. Gaussova metoda je založena na konvoluci za použití vhodného konvolučního jádra, stejně jako průměrování. Hodnoty konvolučního jádra se počítají dle rovnice Gaussovy funkce (1). Gaussova funkce je směrodatná odchylka  $\sigma$ , která určuje poloměr rozostření v jednotlivých pixelech. Použijeme-li jednotlivá konvoluční jádra 3x3 a 4x4 výsledek bude stejný, jako při použití jednoho konvolučního jádra 5x5, jelikož výsledek je odmocnina součtu druhých mocnin poloměrů.

$$G(x, y) = \frac{1}{2\pi\sigma^2} * e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

1 Gaussova filtrační funkce

### 3.2.2 První derivace Gaussovy funkce

Po provedené filtraci šumu v obrazu za pomoci Gaussovy funkce, je zapotřebí nalézt první derivaci této funkce neboli určit gradient (2). Tato metoda provádí výpočet gradientu obrazu tak, že nejprve vypočítá potřebnou derivaci funkce a následně ji použije pro konvoluci se vstupním obrazem. Je tak provedena operace derivace a konvoluce v jednom kroku.

$$G'(x) = \left(\frac{x}{\sigma^2}\right) * e^{\frac{-(x^2)}{2\sigma^2}}$$

2 Gaussova první derivace

### 3.2.3 Nalezení lokálních maxim

Při hledání lokálních maxim dochází k potlačení hodnot, které nejsou lokálními maximy. Dojde tak ke ztenčení hran a vytvoření jednoznačné odezvy na hranu v obraze. Algoritmus pracuje na principu hledání a výběru lokálních maxim z hodnot gradientů. Nelze-li v daném pixelu nalézt lokální maximum je tento pixel označen za nehranový a eliminován. Vyjdou tedy na povrch pouze pixely hranové (osahující lokální maximum).

### 3.2.4 Eliminace nevýznamných hran

Proces eliminace nevýznamných hran je část zpracování obrazu, kdy je gradientní obraz převeden do binární podoby. Na takovém obrazu je provedeno prahování či segmentace prahování (3). Rozlišujeme dva typy prahování: globální prahování a prahování s hysterezí.

$$g(i, j) = 1 \text{ pro } f(i, j) > T$$

$$g(i, j) = 0 \text{ pro } f(i, j) \leq T$$

**3.2.4.1 Globální prahování (4)**

Globální prahování pracuje na principu histogramu, což znamená absolutní četnost zastoupených hodnot v obraze. Četnost výskytu daných hodnot znamená, kolik pixelů v obraze obsahuje danou hodnotu intenzity. Výsledkem metody je průměrná hodnota intenzity daného obrazu, například hodnota 60.

$$T = T(f)$$

Globální prahování znamená, že práh je určen z oblasti celého snímku. Na rozdíl od lokálního prahování, při kterém určíme práh v závislosti na pozici v obraze.

**3.2.4.2 Prahování s hysterezí (5)**

Tento způsob prahování pracuje se dvěma krajními body výsledků, což znamená horní mez hystereze a dolní mez hystereze. Samotný výpočet jednotlivých hodnot je přiřazování intenzity jednotlivým hodnotám. Pokud je porovnávaná hodnota vyšší, než maximální hodnota prahování je porovnávaná hodnota považována za novou maximální hodnotu prahování (255). Pokud je porovnávaná hodnota nižší, než minimální hodnota prahování je označena za nevýznamnou a je jí přiřazena hodnota 0. Proto je výsledkem této metody oblast hodnot (interval), například 125 až 78.

$$T \in [T_L; T_H] \cap T > T_H$$



## 4 Popis používaných sběrnic při praktickém návrhu SW ovladače pro zařízení HSX

### 4.1 HSX

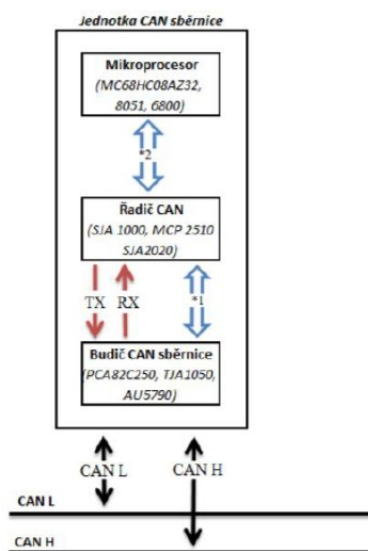
HSX je zkratka diagnostických rozhraní řady HS, které jsou používány společně s komunikačním SW a standardizovaným rozhraním D-PDU API v souladu s ISO 22900-2. Spojení umožňuje integraci do D-serveru a nepřímou integraci HS rozhraní do testovací aplikace (SW Framework). Rozhraní HS je připojeno pomocí multi-linkové komunikace k patřičným ŘJ vozidla. Díky propojení lze při výrobním procesu testovat zařízení připojená k jednotce a umožnit diagnostiku a flashování pomocí testovací aplikace. Rychlost odezvy a komunikace je vysoká, a to díky protokolu HS rozhraní, který pracuje na výkonném 32bitovém procesoru.

### 4.2 Sběrnice CAN

Sběrnice CAN je sériová sběrnice vyvinutá firmou Bosch pro přenos dat. Sběrnice je dvou vodičový uzavřený obvod, který dle potřeby může být stíněný či nestíněný. Obvod obsahuje na obou koncích vedení rezistory o hodnotách 120 Ω. Parametry přenosu vychází z normy ISO 11898. Sběrnice CAN je multi-master sběrnicí, což znamená, že každé zařízení na sběrnici může zprávy vysílat i přijímat (naslouchat). Protokol CAN odpovídá datové vrstvě přenosu ISO/OSI modelu. Síťová vrstva detekuje a opravuje chyby přenosu vzniklé na sběrnici vlivem okolních elektromagnetických polí. Podporuje centrální diagnostiku sběrnice a snadnou konfiguraci.

Data neobsahují adresu příjemce ale identifikátor přenosu, který je jedinečný a definuje obsah přenášené zprávy a její prioritu. Všechna zařízení na sběrnici mohou naslouchat a sama vyhodnotí, zda je zpráva určena pro ně. Maximální rychlost na sběrnici je 1Mbit/sec.

CAN protokol podporuje dva rámcové formáty zpráv. Podstatný rozdíl je v délce označení (ID). Ve standardním formátu je délka ID 11 bitů a v prodlouženém formátu je délka označení zprávy 29 bitů.[5]

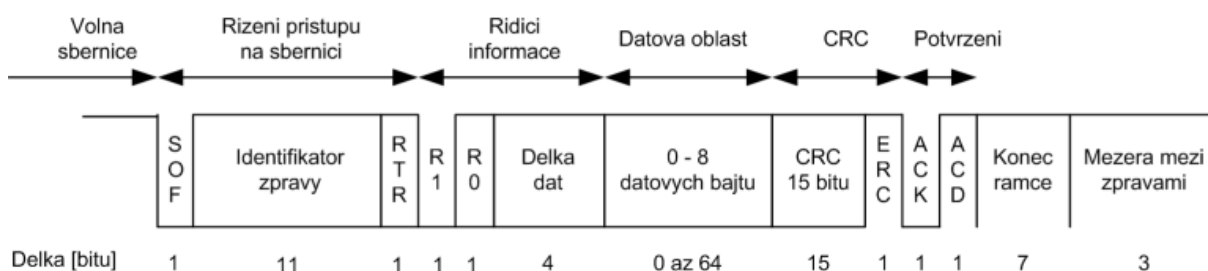


Obrázek 20 Připojení jednotky na sběrnici CAN [5]



### 4.2.1 Typy zpráv CAN protokolu

1. Data Frame (Datová zpráva) – datová zpráva
2. Remote Frame (Žádost o data) – stejný formát jako datová zpráva, RTR bit nastaven jako recessive, chybí datová oblast rámce
3. Error Frame (Chybová zpráva) – slouží pro signalizaci chyb na sběrnici CAN, posílána od kteréhokoliv uzlu, který detekuje chybu na sběrnici
4. Overload Frame (Zpráva o přetížení) – slouží pro oddálení či pauze mezi vysláním další zprávy nebo žádosti o data



Obrázek 21 Datový rámec CAN sběrnice [5]

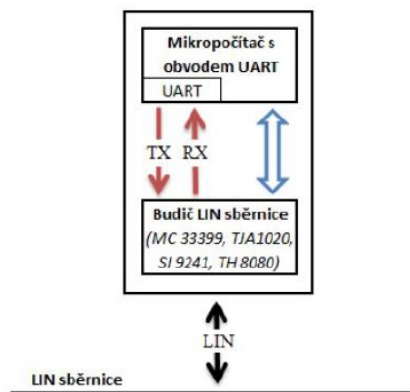
### 4.2.2 Popis datového rámce zpráv

1. Start bit – počáteční bit
2. Arbitrační bit – identifikátor, rozhodovací pole pro přístup na sběrnici (11 bitů nebo 29 bitů)
3. RTR bit – žádost o vzdálený přenos, ukazuje, zda jde o rámec dat nebo požadavek rámce bez bitů o velikosti 1 bit
4. Kontrolní pole – obsahuje IDE 1 bit, jedná se o označení, zda se jedná o klasický či prodloužený formát
5. Rezervní bit + DLC – DLC je výpočet 4 bitů z bytů dat v datovém poli
6. Datové pole – jedná se o velikost pole 0 až 64 bitů samotné zprávy
7. Kontrolní součet – bezpečnostní kontrola rámce na detekci bitových chyb o velikosti 15 bitů
8. ACK – vratný (přepisovaný) bit potvrzení správného přijetí zprávy od příjemců, je přepisován jako dominant bit
9. Konec rámce – celkem 10 bitů, konec posílaného rámce + přerušení (mezera), které odděluje následné zprávy, pokud se nějaké vyskytnou, pokud ne zůstává nečinné

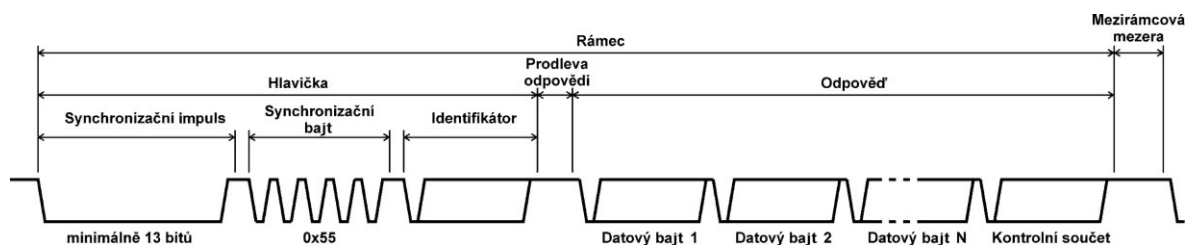
## 4.3 Sběrnice LIN

Sběrnice LIN je sériová asynchronní sběrnice, tedy jedno-vodičové spojení připojených zařízení. Byla navržena pro použití v automobilové technice s ohledem na minimální cenové náklady. Jejím cílem není nahradit sběrnici CAN, ale doplnit ji tam, kde není vyžadován tak přesný přenos, nebo zde není zapotřebí použití drahé sběrnice pro komunikaci.

Jedná se o typ sběrnice single-master/multiple-slave, kde jedno zařízení kontroluje a řídí komunikaci (provádí synchronizaci) a ostatní podřízená zařízení s ním komunikují. Koncepte budičů sběrnice vychází ze standardu ISO 9141 s vylepšeními v oblasti EMC. Maximální rychlost sběrnice 40kbit/sec. [6]



Obrázek 22 Připojení jednotky na sběrnici LIN [5]

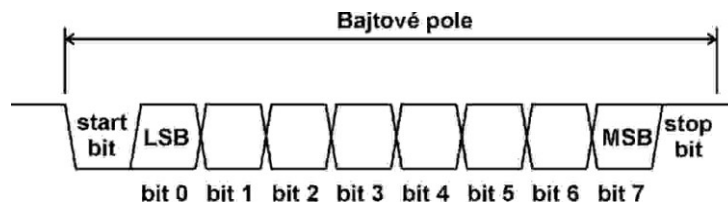


Obrázek 23 Datový rámec sběrnice LIN [6]

#### 4.3.1 Popis datového rámce zprávy

Hlavička – synchronizační impuls (vysílá pouze master zařízení), synchronizační bajt a identifikátor

Odpověď – bajtové pole dat, včetně start bitu a stop bitu



Obrázek 24 Základní datové pole zprávy [6]

#### 4.3.2 LIN řídicí zařízení (Master)

Nadřazené zařízení má na starosti:

- Definuje přenosovou rychlost
- Vysílá synchronizační impuls, synchronizační pole, identifikátor
- Monitoruje a potvrzuje data pomocí kontrolního součtu
- Přepíná podřízené jednotky do sleep módu a znovu je probouzí
- reaguje na signál probuzení od podřízených jednotek.

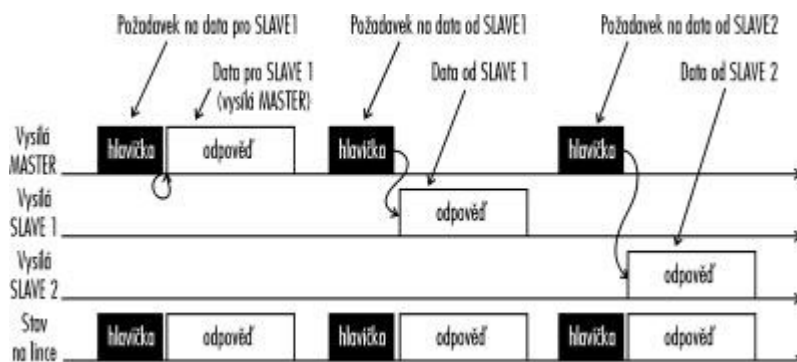
#### 4.3.3 LIN podřízené zařízení (Slave)

Podřízené zařízení má na starosti:

- Čekání na synchronizační impuls
- Synchronizuje se podle synchronizačního pole

Podle identifikátoru dělá jednu z následujících činností:

- Nereaguje
- Přijímá data
- Vysílá data
- Kontroluje nebo posílá kontrolní součet



Obrázek 25 Řízení přístupu zařízení na sběrnici LIN [6]

Touto kapitolou končí teoretický rozbor a následuje popis mého vlastního díla.

## 5 Teoretický rozbor a návrh řešení kamerové inspekce průmyslového výrobku

V kapitole teoretického rozboru a návrhu řešení kamerové inspekce průmyslového výrobku je popsána praktická část výběru komponent pro optické testy a studie proveditelnosti.

Při zadání požadavků ze strany zákazníka dochází zhotovitelem ke zpracování a seskupení požadovaných optických kontrol. Zákazník vytvoří dokument s přesně popsány částmi kontroly na vzorovém příkladu výrobku. Následně je vytvořený dokument označen za zadávací specifikaci a na straně zhotovitele je provedena studie proveditelnosti. Při této studii dochází k vytipování vhodných komponent pro úspěšnou optickou kontrolu artefaktů výrobku. Je obvyklé, že se testuje několik komponent, například různé typy kamer: monochromatické či polychromatické; testování objektivů různých parametrů vzhledem ke vzdálenosti objektu a kamery; různé osvětlovače pro nasvícení scény a zviditelnění hledaných částí. Takto provedená studie obsahuje parametry ohledně vytipovaných komponent, detailní popis sestavy použité při studii a několik snímků s různými komponentami. Obsahuje jednoduše vyznačené oblasti nalezených artefaktů. Cílem studie proveditelnosti je vytvořit návrh detekce oblastí výrobku včetně použitých komponent a stručně popsaného postupu detekce částí výrobku.

Takto provedená studie je předána vedoucímu projektu, který provede nacenění zakázky. Po potvrzení zájmu a odsouhlasení zákazníkem provede vedoucí projektu rozdělení práce na vývoji testeru. Při vývoji SW komponenty pro zadanou problematiku jsou použity již vytvořené snímky ze studie proveditelnosti. Na konci vývoje probíhá testování úspěšnosti nalezených artefaktů a doladění potřebných částí.[10]

### 5.1 Návrhu řešení kamerové inspekce pro kontrolu skla automobilu

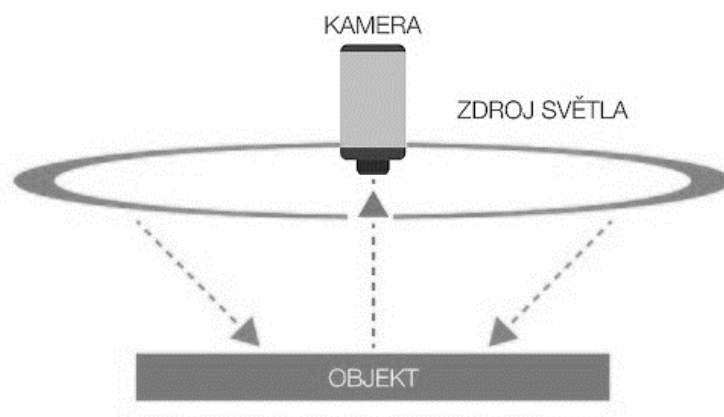
Zákazníkem bylo požadováno detekování bočního skla automobilu ve fázi výstupní kontroly výrobku. Při detekci skla bylo kontrolováno množství nanesené pasty po obvodu skla, která je zde zapotřebí pro upevnění lepidla při montáži skla na automobil. Teplo-vodivá pasta, jež byla nanášena robotem, byla kontrolována ihned po nanesení, jelikož její vystavení UV záření způsobuje schnutí a snižování barevné odlišnosti vůči ostatním částem skla. (Obrázek 27)

#### 5.1.1 Použité komponenty

- Kamera Basler ace 2440–20gm
- Objektiv Kowa ohnisková vzdálenost 5 mm LM5JC10M
- Osvětlovač ATE Light 900x600mm plochý, napájení 24VDC

#### 5.1.2 Studie proveditelnosti

Při studii bylo pracováno s více teoriemi nasvícení. Výsledná volba umístění osvětlovače byla ta, že osvětlovač byl umístěn nad samotným sklem. Kamera byla umístěna ve středu osvětlovače, který byl pro tento systém přizpůsoben otvorem pro kameru. Další teorie byla, že se scéna nasvítí z boční strany pro zviditelnění povrchu čili zviditelnění nanesené pasty, jenž vyčnívá z povrchu skla.



Obrázek 26 Umístění komponent kamerové sestavy – temné pole [4]

Po testování a pořízení několika snímků bylo zjištěno, že ideální umístění osvětlovače je horní pozice (Obrázek 26). Kamera, jak již bylo napsáno, se nachází uvnitř osvětlovače, v jeho středu.



Obrázek 27 Snímek skla při studii

### 5.1.3 Kamerová sestava, nainstalována v průmyslové výrobě

Kamera je umístěna ve středu osvětlovače, který nasvĕcuje scĕnu z horní části (Obrázek 28). Osvĕtlovač je triggrován pomocí kamery, která si sama kontroluje jeho rozsvícení.



Obrázek 28 Sestava inspekce pro detekci teplo-vodivé pasty na obvodu skla

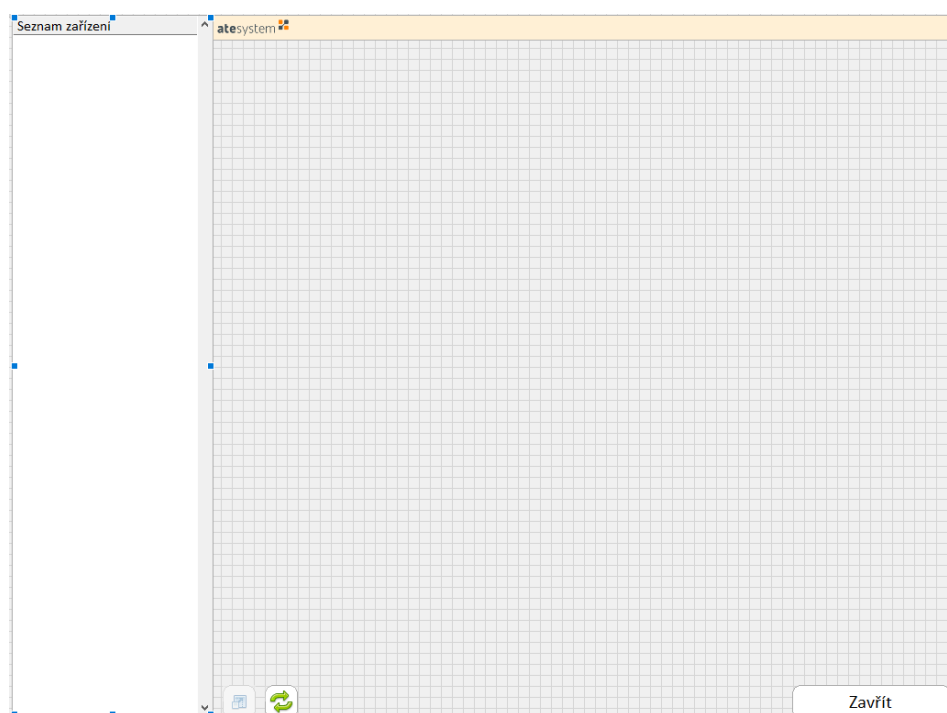
## 6 Návrh uživatelských rozhraní SW

V této kapitole je popsána vlastní činnost při návrhu či úpravě uživatelských rozhraní ve firemním SW Frameworku (dále jen SW Framework). SW Framework slouží pro aplikace strojového vidění. Komunikuje s připojenými zařízeními pomocí implementovaných SW ovladačů, obsahuje SW moduly, které provádí jednotlivé testy, a spojuje jednotlivé části do ucelené aplikace nasazené v průmyslové výrobě. Aplikace obsahuje definiční soubor pro nastavení jejího chování v operačním systému Windows. Řídící část SW frameworku testovací aplikace se nazývá jádro, které řídí sekvenci volání jednotlivých částí pro zpracování dat měření, snímání obrazu či komunikaci se zařízením. Jádro obsahuje čelní panel, který však není viditelný v aplikaci, panely aplikace jsou totiž složeny z příslušných panelů SW modulů a SW ovladačů. Sestavení sekvence testů, nastavení limitů či výstupů v jednotlivých fázích je prováděno ve skriptovacím souboru jádra. Jádro systému je samostatně spustitelné, a pro svůj chod nepotřebuje žádné další moduly či zařízení, avšak pro samotné testování výrobků je nepoužitelné. Z toho vyplývá, že SW ovladače pro komunikaci se zařízeními a SW moduly pro zpracování měřených dat nebo vizualizaci obrazu z kamery jsou nedílnou součástí celé aplikace v procesu optické kontroly průmyslového výrobku. Cílem vývoje optického testeru pro konkrétní účel je složit správnou sekvenci a následně připojit k jádru všechny potřebné SW části tvořící jako celek aplikaci pro optickou kontrolu výrobku v průmyslové výrobě.

Uživatelská rozhraní reprezentují části, které lze vizualizovat. Dále provádí reakce na změny hodnot prvků, vizualizace fází procesu nebo aktuální stav či místo, kde se aplikace nachází. SW Framework je postaven na principu obrazového náhledu z kamery, tudíž hlavní část obrazovky zabírá panel náhledu kamery či snímáný obraz obsahující označení zajímavých nebo kontrolovaných částí výrobku. V pravé části se nachází tabulka reportů, která se pravidelně aktualizuje a zobrazuje tak aktuální fázi procesu aplikace. Naměřené hodnoty testů a jejich úspěšnost spolu s časovou značkou informují o výsledku a době trvání testu, podbarvení názvu testu v tabulce reportů označuje jeho úspěšné vykonání a výskyt chyby. Ve spodní části obrazovky jsou ovládací prvky pro obsluhu stroje. Tyto prvky slouží pro vstup do menu aplikace nebo nastavení konfigurace.

### 6.1 Popis rozhraní SW Frameworku pro testování

Uživatelské rozhraní aplikace je rozvětveno na menší části. Díky tomuto rozvětvení jsou jakékoliv úpravy jednodušší a řeší se vždy příslušná část. Jako celek však musí být dodržena vizualizace, návaznost a jednotná vzhledová definice. Vývoj SW komponent je tedy v hierarchii SW Frameworku jedním z nejdůležitějších prvků. Vývoj je rozdělen na dvě fáze, nejprve se vždy řeší samotný algoritmus popsatelný pomocí vývojových diagramů či UML diagramu. Algoritmus reprezentuje implementovanou část komponenty. Tato část je uživatelsky nepřístupná, je rozdělena do několika částí a stará se o změny prvků čelního panelu, reakci na tlačítka a o samotnou funkci SW komponenty, pro kterou byla vytvořena. Druhá fáze vývoje je tvorba čelního panelu (Obrázek 29), tedy uživatelského rozhraní SW komponenty. Vizualizace prvků či jejich rozmístění vychází z firemního předpisu. Většinou se na čelním panelu nachází pouze prvky, které je potřeba ovládat či měnit jejich hodnoty pro správný proces testu. Části, které nepodléhají častým změnám parametrů, avšak jsou podstatné pro vyhodnocení algoritmu jsou definovány v příslušném souboru s příponou .ini. Díky této struktuře se podstatně zjednoduší obsluha celé aplikace a možnost tak nezávisle přidávat další SW moduly či SW ovladače do stávající aplikace.



Obrázek 29 GUI pro nový SW ovladač

Vytvořené GUI pro SW ovladač (Obrázek 29) neobsahuje žádné specifikované prvky pro daný typ SW ovladače. Lze tedy tento návrh použít jako čistý SW ovladač při integraci zařízení. Díky této přípravě lze ihned přistoupit k programování kódu a dalšímu vývoji SW ovladače podle specifikací zařízení.

## 6.2 Postup při návrhu GUI

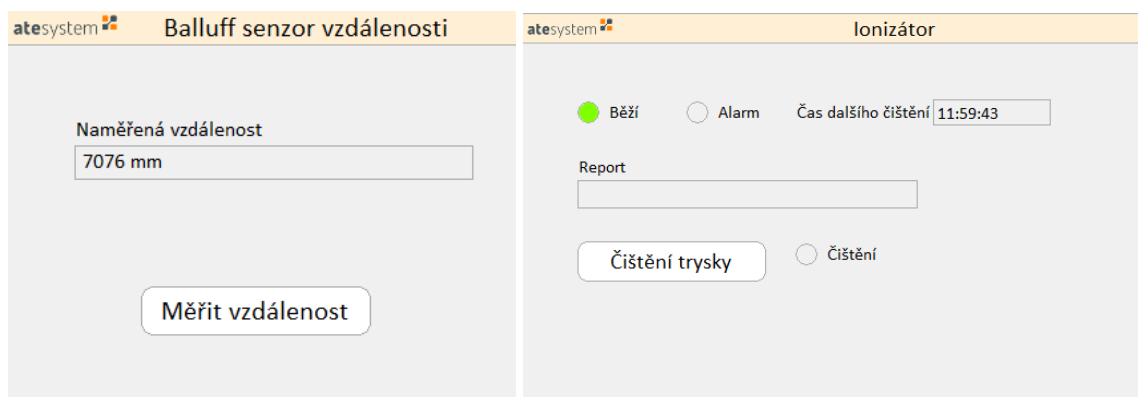
Při návrhu uživatelských rozhraní se většinou postupuje podle předem definovaných kroků vytvořených v závislosti na komplexnosti řešení, ale také dodržení firemního standardu pro tvorbu aplikace. Při tvorbě nové aplikace je potřeba dodržet firemní standard tak, aby dvě odlišné aplikace měly jednotné GUI, a navenek tak reprezentovaly kvalitní zpracování SW Frameworku.

Vytvoření soupisu funkcí, pro které je aplikace vytvořena je zásadní pro následující vývoj GUI SW komponent. Pokud má být aplikace pouze informativního charakteru a uživatel či operátor nemá vstupovat do nastavení nebo parametrizace je doporučeno zvolit přehledný a velmi jednoduchý vzhled. Pokud SW komponenta obsahuje části, které upravují nastavení běhu či nastavení parametrů pro správné vyhodnocení, je zapotřebí doplnit tuto část o sekci parametrů, které se zobrazí na čelním panelu. Cílem je vytvořit jednotný návrh uživatelského okna, který však bude pro každou SW komponentu zvlášť modifikovatelný (tvorba, tlačítek, indikátorů, obrázků nebo grafů). Návrh takového GUI aplikace je vytvořen jako okno aplikace s neměnnými tlačítky pro nastavení, ukončení a změnu konfigurace aplikace. Mění se pouze vnořená část, která například na hlavním panelu zobrazuje obrázek z kamery na konci testu a tabulku výsledků testu. Na ostatních panelech je část panelu, kde je zobrazen náhled snímku, je definována podle specifické funkce panelu. Panel nastavení bude obsahovat několik numerických či textových prvků pro parametrizaci, panel pro ovládání kamery bude obsahovat aktuální náhled kamery a tlačítka pro pořízení snímku či pro trvalé snímání obrazu. Upřesňující nastavení kamery týkající se expozičního času či dalších parametrů budou nastavovány ze souboru nastavení s příponou .ini.



Vyskakovací okna aplikace jsou rovněž tvořena podle předpisu GUI. Tato okna se většinou mění podstatně méně, a proto lze vytvořit několik dialogových oken a ty později přenášet mezi aplikacemi dle potřeby. Mezi neměnná okna patří dialog pro ukončení aplikace, dialog pro potvrzení uložení změny konfigurace nebo dialog hlásící chybu měření, ve kterém se bude měnit pouze místo informace o výskytu chyby a číslo chyby dané aplikace.

Hlavní požadavky ze strany zákazníka jsou vytvoření fungujícího celku, intuitivní chování a zcela jasně definované parametry pro úspěšnou konfiguraci testu. Vytvořit GUI aplikace, které s dalším vývojem bude neměnné, to znamená, že při nasazení nových SW komponent bude princip a vzhled aplikace stálý. Související požadavek je také distribuce aplikací ve firmě zákazníka, pokud se porovná několik firemních aplikací různého určení, měly by vzhledově působit jednotně a přehledně. Například aplikace pro ovládání proudového zdroje a hlavní aplikace SW frameworku testovací aplikace by měly obsahovat vzhledově stejné prvky uživatelského panelu, stejný vzhled grafů i stejné vizualizace výsledků či měřených hodnot v podobě tabulky reportů (Obrázek 30).



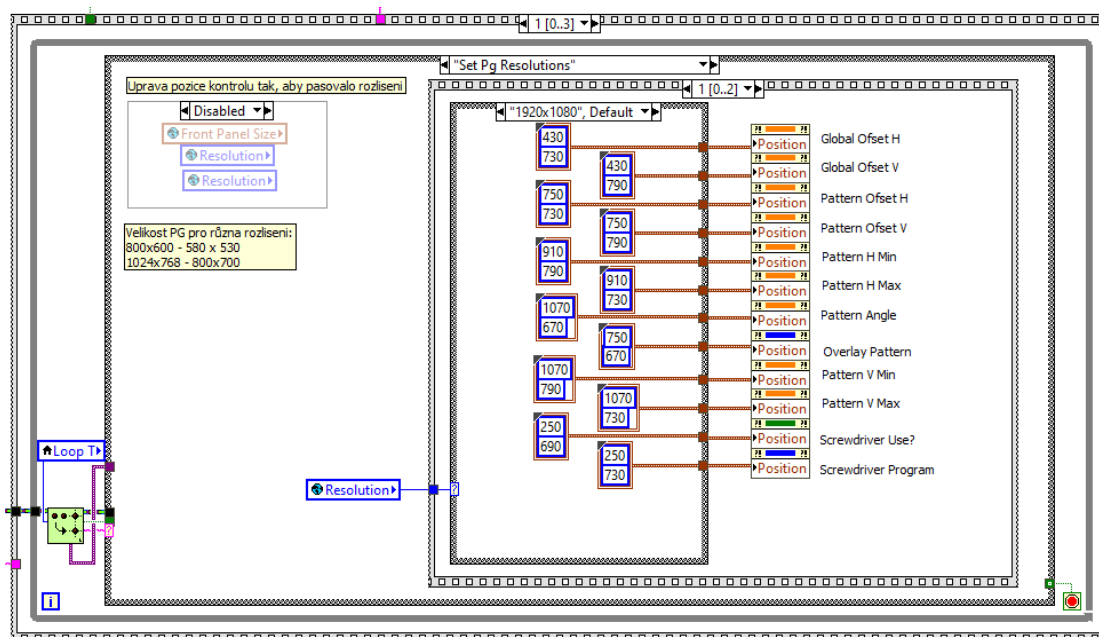
Obrázek 30 Srovnání dvou odlišných SW ovladačů se stejným GUI

### 6.3 Rozložení prvků panelu a interpretace popisků

Rozložení prvků na přesně danou pozici čelního panelu úzce souvisí s vytvářením GUI. Jelikož máme několik možností rozlišení, chceme zobrazovat jen některé prvky či navázat jejich viditelnost na souběh událostí programu. Potřeba definovat pozici a uzly vlastností prvků čelního panelu je proto nezbytné. Pokud vytváříme SW komponenty pro použití v aplikaci, měly by tyto komponenty na první pohled vypadat totožně. Díky možnostem prostředí LabVIEW lze veškeré chování prvků nastavit.

Pokud potřebujeme měnit jazykový překlad programu, je možné navázat popisky prvků právě na zvolený typ jazyka, a tím zajistíme vždy správné zobrazení jazyka. Veškeré tyto změny se provádí přes vlastnosti prvku a poté zápis jejich nové hodnoty v určité části programu.

Při rozložení prvků volíme vlastnosti pozice, ve které je prováděno nastavení vzdálenosti prvku od levého horního rohu obrazovky. A máme tak jistotu, že se prvky nebudou překrývat a nebude docházet ke kolizi.



Obrázek 31 Část obsahující nastavení pozice prvků čelního panelu podle zvoleného rozlišení

## 7 Návrh statistického zpracování výsledných dat

Kapitola návrhu statistického zpracování výsledných dat se věnuje popisu praktické části diplomové práce, při které byl realizován návrh zpracování dat z průběhu měření. Zpracování dat je nedílnou součástí měření. Získávání jednotlivých výsledků měření a jejich uložení umožňuje následné zpracování statistických parametrů vyhodnocení, porovnání naměřených hodnot během směny či odhalit nedostatky při výrobě součástky. Umožňuje tak zákazníkovi nahlédnout do průběhu jednotlivých měření celkového testu a celkem spolehlivě zjistit příčinu problému.

Prvním důvodem pro tvorbu reportu dat je statistické zpracování dat z výroby za určitý časový úsek. Většinou se jedná o měsíční rekapitulaci vyrobených kusů a kontrola počtu OK a NOK kusů. Dalším důvodem může být report měření odesílaný na úložiště. Report většinou slouží k pozdější kontrole, zda operátor prováděl veškeré úkony ve správném pořadí, nebo zda nedošlo k poškození zařízení a měření nebylo ovlivněno vnějšími vlivy ze strany operátora nebo poruchy celé linky.

### 7.1 Naměřená data mohou být uložena do souboru s příponou .txt

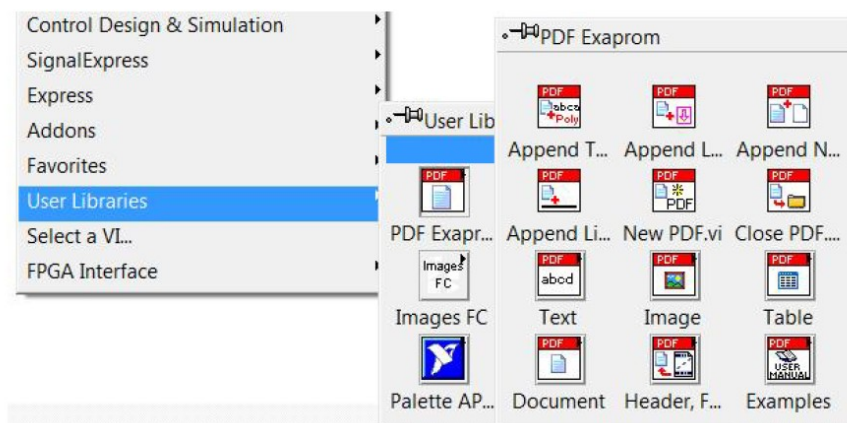
Soubor s příponou .txt je velmi častý typ souboru pro ukládání naměřených hodnot nebo ukládání logovacích souborů obsahujících informace o úspěšně provedené sekvenci testů celé aplikace. Díky jednoduchým knihovnám pro práci s textovými soubory lze jednoduše navrhnout jejich strukturu, tedy hlavičku souboru, identifikátor testovaného kusu (sériové číslo výrobku či specifické označení) a jednotlivé sekce, ve kterých se mění hodnoty z jednotlivých měření. Textové soubory jsou však méně přehledné v porovnání s ostatními typy. Většinou obsahují mnoho dat a nalézt v několika řádkovém souboru chybu je velmi obtížné.

Ukončení měření v aplikaci spočívá ve vyhodnocení měřeného kusu jako OK či NOK kus. Následně je zavolán modul, který má na starost sběr dat. Modul je volán samotným jádrem ihned po vyhodnocení kusu, obdrží naměřené hodnoty a identifikátor vyrobeného kusu. Tyto parametry následně uloží do souboru s příponou .txt. Název tohoto souboru se musí lišit mezi jednotlivými kusy, je proto generován dynamicky pomocí časové značky a sériového čísla přčteného z výrobku pomocí čtečky kódu nebo kamery. Vytvořené soubory jsou uloženy do příslušné složky většinou pod označením data. Přístup k této složce a k jednotlivým souborům není omezen a je umístěna na stejném místě jako ostatní podstatné části aplikace.

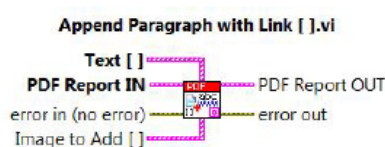
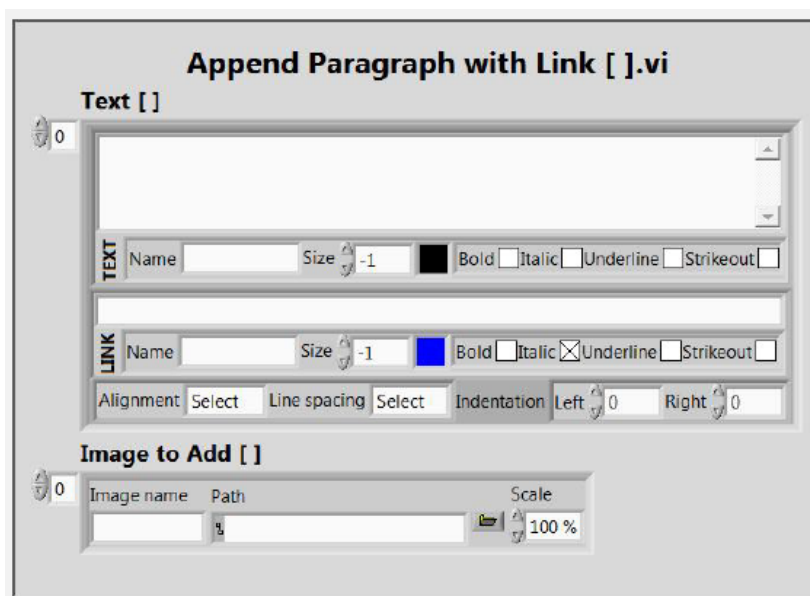
### 7.2 Naměřená data mohou být uložena do souboru s příponou .pdf

Pro práci s tímto typem souboru je zapotřebí doinstalovat speciální knihovny vývojového prostředí LabVIEW. Knihovny pro práci s PDF (Obrázek 32 a Obrázek 33) formátem nejsou součástí prvotní instalace aplikace LabVIEW od společnosti NI. Soubory obsahující informace o dokončeném testování výrobku s příponou .pdf jsou velice přehledné a zpravidla se do nich neukládá tolik informací.

Ukončení měření aplikace nemusí vyhodnotit kus jako OK či NOK, ale může být vyhodnocen na základě kombinace hodnot. Pro tento typ je ideální použít soubory s příponou .pdf. Pomocí funkcí lze vytvořit uživatelsky definovaný soubor obsahující grafy, tabulky či textové informace o průběhu a výsledku testu. Tento typ, jak již bylo zmíněno je velice přehledný a porozumí mu i méně znalá osoba, která má na starost kontrolu a zpracování výsledných dat vyrobených kusů.



Obrázek 32 Knihovny pro ukládání souborů \*.pdf [13]

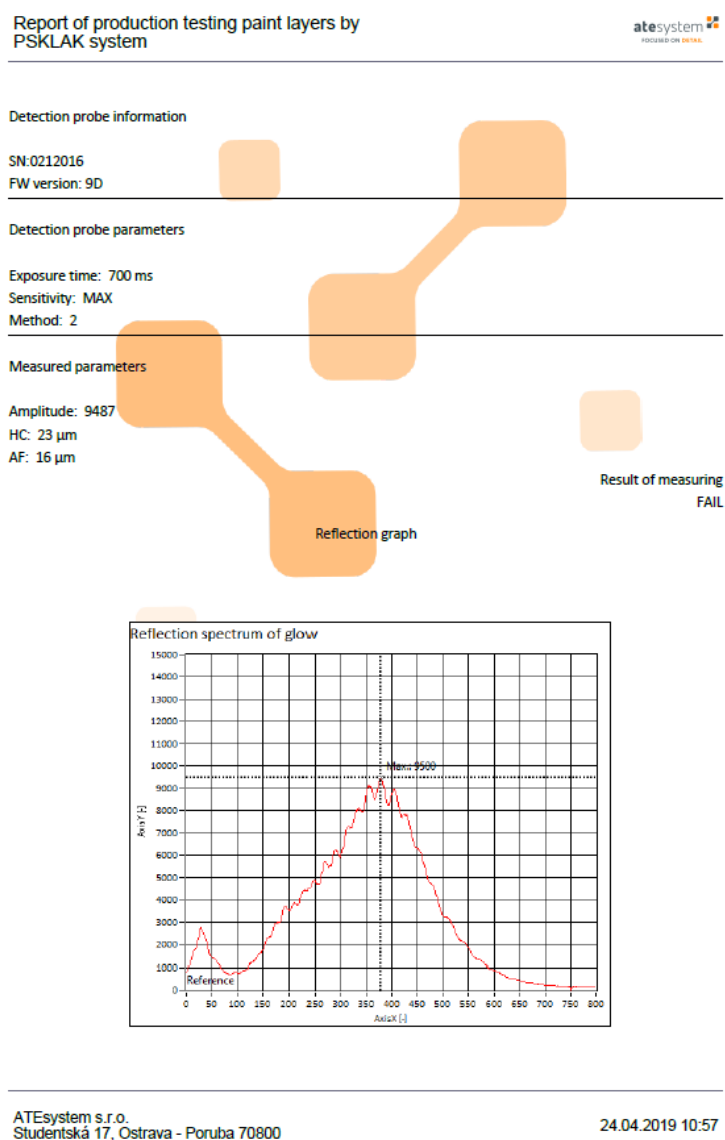


Obrázek 33 Nastavení obsahu a vzhledu textu pomocí knihoven Exaprom PDF [13]

### 7.3 Report měření nanesené tloušťky krycí vrstvy laku na krycím skle světlometu

Cílem bylo vytvořit reportový list obsahující informace o naměřených hodnotách tloušťek lakových vrstev na krycím skle. Vytvořený report měření zaznamenává důležité hodnoty pro výsledek měření a následné nastavení měřícího zařízení spektrometru. PDF report byl vytvořen v aplikaci zabývající se měřením tloušťky lakových vrstev na krycích sklech světlometů (Obrázek 34). Report sloužil jako

přehledná ukázka výsledků testu zákazníkovi. Měření tloušťek lakových vrstev probíhalo na více místech skla světlometu. Díky takto vytvořenému reportu bylo možné upravit výrobní proces nanášení lakové vrstvy tak, aby pokrytí skla lakovou vrstvou bylo rovnoměrné a měřené hodnoty byly ustáleny do úzkého intervalu hodnot. Byly zde nastaveny limity pro splnění testu a jeho označení jako TRUE. Tyto limity korespondovaly s limity vyhodnocení v samotné aplikaci pro měření tloušťky laku spektrometrem.



Obrázek 34 Report měření tloušťky laku s nastavenými parametry

## 8 Vývoj SW komponent v programovacím prostředí LabVIEW pro firemní testovací SW Framework

V následujících podkapitolách je popsán praktický vývoj SW komponent pro sestavení SW Frameworku a nasazení aplikace v průmyslové výrobě. V kapitole 8.1 je popsán vývoj SW modulu kontroly nanesené pasty na skle automobilu. Kapitola 8.2 je věnována popisu vývoje SW ovladače pro komunikaci s ŘJ světlometu automobilu pomocí komunikačního rozhraní, které vytváří zařízení HSX. V následující kapitole 8.3 je popsán vývoj jednotlivých SW ovladačů, které pomocí komunikačního média posílají naměřená data do aplikace pro kontrolu světelné stopy. Poslední kapitola 8.4 je věnována popisu vývoje algoritmu pro vykreslování křivek v SW modulu kamerového sekvencéru, který řídí posloupnost optických testů v aplikaci pro kontrolu světelné stopy.

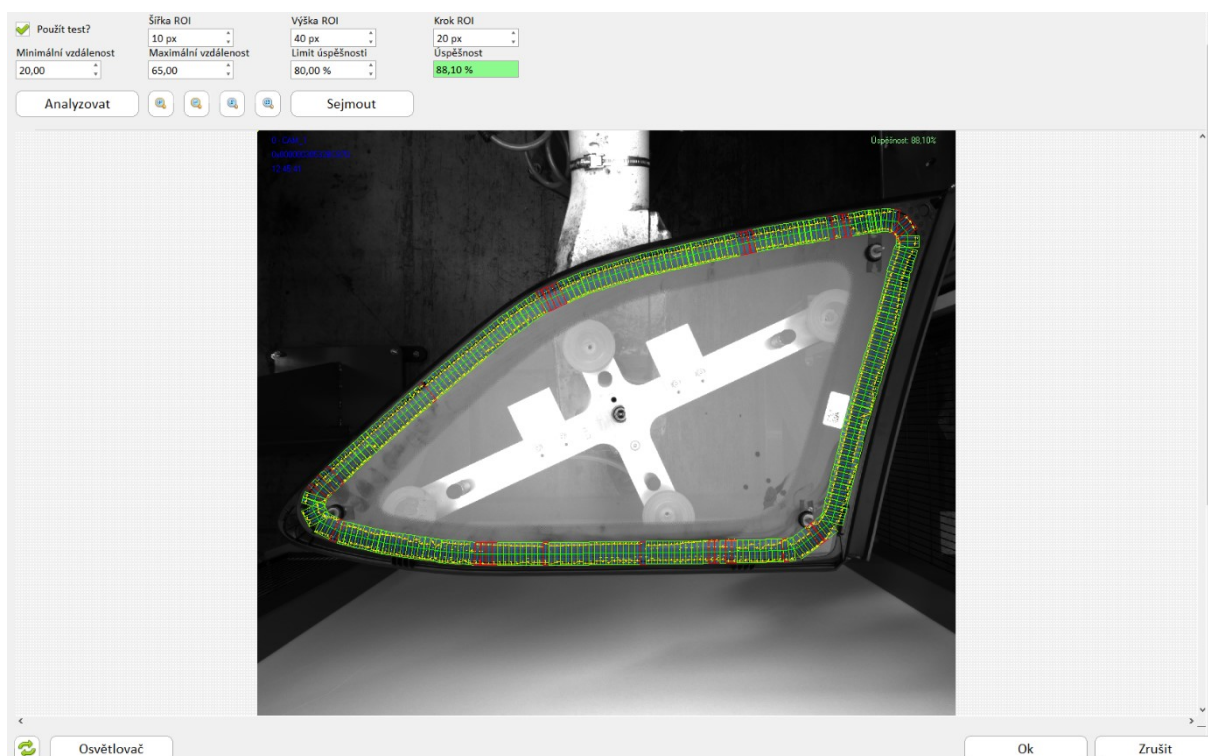
### 8.1 SW modul kontroly nanesené pasty na skle automobilu

SW modul pro kontrolu nanesené pasty slouží pro hledání hran v uživatelsky definované oblasti zájmu. Požadavkem bylo detekovat a sledovat tloušťku naneseného nátěru teplo-vodivé pasty na skle automobilu při výstupní kontrole výroby. Bylo požadováno provést vyhodnocení na základě OK a NOK kusů. Samotný algoritmus zpracování detekce byl zvolen ze zkušenosti definovat si oblast zájmu a poté pomocí funkcí v LabVIEW nasadit hledání hran v této oblasti od zvoleného bodu s definovanými parametry. Parametry pro hledání hran v oblastech zájmu jsou nastavitelné z příslušného souboru SW modulu.

#### 8.1.1 GUI SW modulu pro kontrolu nanesené pasty

Uživatelsky přístupná část SW modulu je čelní panel (Obrázek 35). Zde má uživatel možnost měnit podstatné funkce potřebné pro správné vyhodnocení kontroly, nastavení oblastí detekce a snímat obraz z kamery pomocí tlačítka. Uživatelsky definované prvky jsou šířka, výška a krok oblasti zájmu. Tyto prvky definují velikost oblastí a posun středů těchto oblastí vůči sobě. Zmiňované středy tvoří základ pro sestavení oblasti zájmu (obdélníků), které definují oblast hledání hran v obraze. Dále má možnost uživatel nastavit minimální a maximální vzdálenost dvou hran, což je šířka hledaného nátěru. Robot nanášející teplo-vodivou pastu na sklo automobilu nemá jakoukoliv kontrolu nad tloušťkou provedeného nátěru, proto je tato tloušťka definována jako vzdálenost dvou nalezených hran v definovaném intervalu. Limit úspěšnosti definuje hodnota v procentech, která určuje mez mezi OK a NOK kusem. Pokud algoritmus nalezne hrany v intervalu vzdáleností, v jednotlivých oblastech označí tuto oblast zájmu zeleně, pokud hranu nenalezne nebo není vzdálenost hran v definovaném intervalu, označí oblast zájmu červeně. Konečné vyhodnocení úspěšnosti měření se procentuálně vypočítává na základě zelených oblastí z celkového počtu oblastí zájmu – procentuální úspěšnost.

Tlačítka umožňují uživateli ručně analyzovat daný snímek, sejmout obraz z kamery nebo zapnout osvětlovač. Uživatel musí zvolit trajektorii testování, jedná se o vykreslení několika úseček, které mají vždy společný krajní bod. Trajektorie je zadávána přímo do pořízeného obrazu, operátor vytvoří sekvenci úseček ukončenou dvojklikem. Daná oblast se podbarví modře. Nyní už může být provedeno samotné vyhodnocení pomocí tlačítka Analyzovat. Tímto způsobem se operátor ujistí, že provedl správný výběr trajektorie. Trajektorii lze měnit, a sice posunem jednotlivých bodů do oblasti nátěru, nejlépe do jeho středu.



Obrázek 35 Čelní panel SW modulu kontroly nanesené pasty

### 8.1.2 Implementace kódu SW modulu

Blokový diagram SW modulu byl vytvořen podle firemního standardu návrhu a tvorby SW komponent. Blokový diagram je složen ze dvou hlavních částí: smyčka pro odchyťávání uživatelský zásahů a smyčka pro vizualizaci snímku pořízeného kamerou, nastavení trajektorie a samotnou analýzu algoritmu pro hledání hran v definované oblasti. Jednotlivé kroky určují sekvence vytvořené na základě názvu jednotlivých částí, sekvence jsou skládány do polí v jedné části kódu. V druhé části jsou kroky rozdělovány na dílčí úkony a postupně řešeny. Například při kliknutí na tlačítko analyzovat je nejprve volána událost „Clear Overlay“ – odebrání oblastí zájmů z obrázku, následně událost „Analyze“ – vykreslení oblastí zájmu podle aktuálně definované trajektorie, poté vyhledání hran v této oblasti (Obrázek 36) a reprezentace výsledků do obrázku ve formě barevného podbarvení oblastí spolu s výpisem hodnoty nalezení hran do indikátoru úspěšnosti.

Algoritmus, který je popisován pracuje s vykreslenou trajektorií, která určuje středy oblastí zájmu. Tyto středy jsou od sebe vzdáleny podle definovaného kroku ROI. Jejich natáčení je dáno úhlem úsečky od vodorovné osy. Tento úhel je vypočítáván pro každou oblast zvlášť. Je patrné, že na jedné úsečce se nachází více oblastí zájmu, a to díky zadané vzdálenosti středů dvou oblastí. Tato část kódu vzhledem ke složitosti a rozsáhlosti je zapouzdřena do samostatné části, která je v kódu reprezentována pomocí vytvořené ikony, tato část se stará o natáčení jednotlivých ROI.

Definice hranového detektoru je v tomto případě určena směrem hledání hrany v dané oblasti, počtem nalezených bodů v jedné rovině reprezentující nalezenou hranu. O vyhodnocení, zda lze požadovat daný počet bodů za hranu se stará funkce „Find edge“. Vizualizace směru hledání hrany je realizována také pomocí LabVIEW funkcí, jedná se tedy o vytvoření řetězce s definovanou posloupností. Na začátku samotného řetězce je surový obrázek pořízený kamerou, který prochází několika funkcemi, jenž do něj

vizualizují svoji část zpracování a společně tak vytvářejí interpretaci celého testu. Konečným krokem posloupnosti funkcí je podbarvení oblastí zájmu a matematického vyhodnocení úspěšnosti testu, poté je obraz aktualizován a vykreslen s již vytvořenými vizualizacemi hranového detektoru.

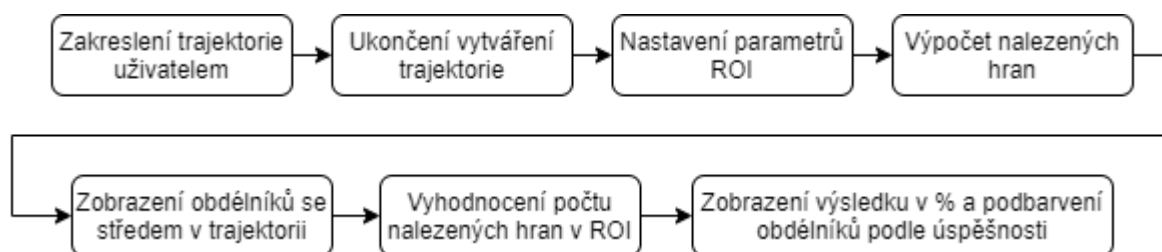
### 8.1.3 Nastavení parametrů SW modulu

SW Framework parametrizuje vývoj SW komponent, a proto každá nově vytvořená komponenta obsahuje stejné části jako ostatní komponenty. Parametry pro úpravu testu SW modulu jsou vloženy do textového souboru s příponou .ini. Tento soubor obsahuje sekce, které definují části parametrizace hodnot.

Rozdělení těchto sekcí je následující:

- Sekce general – všeobecné informace pluginu
- Sekce default – základní nastavení pluginu, definice kamery
- Sekce edge-detector – parametry potřebné k hledání hran

Každá sekce obsahuje několik parametrů, všechny tyto parametry jsou vyčítány při spuštění firemního testeru obsahující daný SW modul. Jsou uloženy do proměnných nesoucí určitou hodnotu na počátku testu. Dochází tak ke zjednodušení změny parametrů přístupných v .ini souboru. Umožňují provést změnu i člověku, který docela dobře nechápe souvislost naprogramovaných částí kódu, ale dobré pojmenování parametrů mu umožní jejich nastavení dle potřeby.



Obrázek 36 Blokový diagram kroků daného algoritmu

## 8.2 SW ovladač pro komunikaci se zařízením HSX

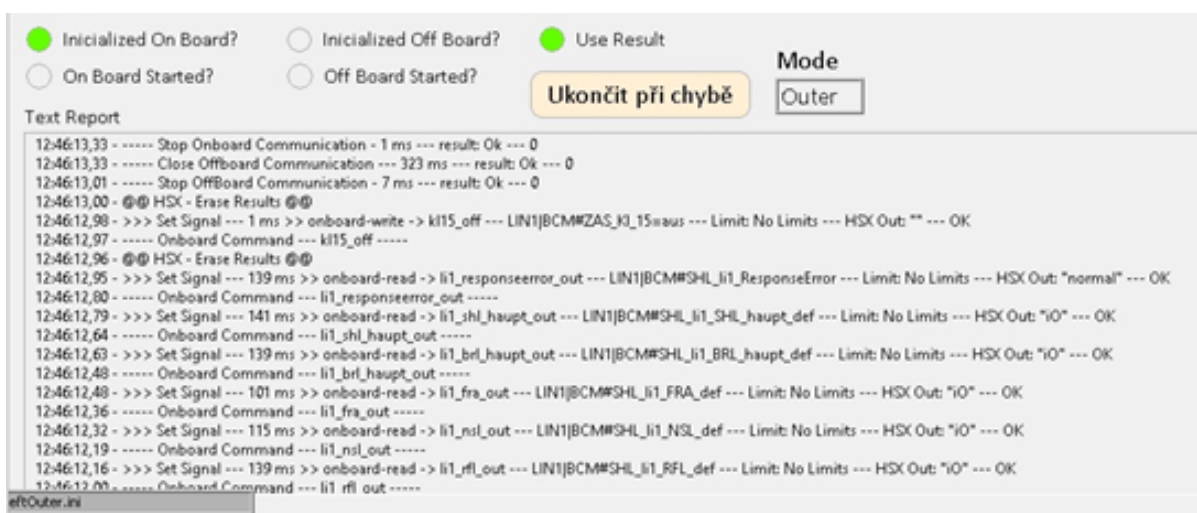
SW ovladač pro komunikaci se zařízením HSX slouží pro komunikaci s fyzickým zařízením HSX jakožto komunikační interface mezi světlometem automobilu a řídicím systémem, kterým je aplikace v PC. HSX v tomto řetězci vytváří druhý koncový bod sběrnice, tedy komunikační rozhraní světlomet – HSX. Data vstupující do počítače jsou přenášena po komunikačním médiu Ethernet. Počítač přistupuje k HSX Serveru a odesílá a přijímá příkazy právě od a do tohoto nadřazeného serveru. Server data přetransformuje do podoby potřebné ke zpracování v HSX zařízení, a ty jsou zařízením posílány jako data ve zprávách standardu automobilových sběrnic (CAN, LIN, K-Line) přímo do jednotky světlometu. HSX zařízení simuluje řídicí jednotku automobilu potřebnou pro spuštění komunikace mezi nadřazeným systémem řídicí části a připojenými zařízeními (jednotka světlometu). Práce s HSX zařízením se využívá pro modifikaci parametrů světlometu a jeho tovární nastavení, čímž je myšlena zvolená strana, výrobní trh, parametry výbavy. Část testování komunikace řídicí jednotky světlometu se nachází při procesu výroby ve fázi parametrizace a testování elektroniky světlometu nebo jeho částí, ale také při výstupní kontrole vyrobeného kusu světlometu a nastavení světelné stopy. Tato část je důležitá pro správnou funkci světlometu v provozu.



Pro správnou funkčnost komunikace je zapotřebí instalace několika přidružených aplikací od společnosti Softing. Aplikace Connection assistant slouží pro nastavení HSX zařízení (IP adresa, porty, komunikační médium) a aplikace System configuration pro přístup k jednotlivým vodičům vedení a pro vložení projektu, který je vytvořen zákazníkem a zadavatelem zakázky. Projekt obsahuje přiřazení vodičů k pinům konektoru na straně světlometu, soubory pro komunikaci po těchto vodičích a dále soubory pro přístup na sběrnici.

### 8.2.1 GUI SW ovladače

Čelní panel zařízení (Obrázek 37) slouží jako indikátor stavů fyzického zařízení HSX. Indikátory reprezentují typ spuštěné komunikační úrovně po sběrnici CAN a LIN. Na panelu se nachází textový indikátor, do kterého je neustále zapisován report komunikace. Jedná se o složení několika informací seskupených v jednom řádku. Podstatné informace pro odhalení vady jednotky světlometu jsou: časová značka odeslání příkazu, typ příkazu, název příkazu reprezentující vlastnost příkazu (pod příkazem se skrývá řetězec číselných dat v hexadecimální podobě), nastavené limity a příchozí odpověď z ŘJ světlometu (nejčastěji OK, No Error).



Obrázek 37 SW ovladač zařízení pro komunikaci d ŘJ světlometu

### 8.2.2 Implementace kódu SW ovladače

Implementace kódu pro vývoj SW ovladače byla rozdělena do několika fází. Prvotní fáze navázání komunikace s ŘJ světlometu provolává sekvence příkazů pro ověření údržnosti této komunikace i při chybně volaném příkazu či absenci odpovědi ŘJ na dotaz. Úkolem bylo nalézt řešení ve formě nadřazené aplikace pro komunikaci a vytvoření tak SW ovladače použitelného pro další typy ŘJ a modulů světlometů. Příkazy byly obdrženy ve formě zákaznické specifikace spolu s přesně požadovanou časovou odezvou a trváním dotazování. Vzhledem k tomuto zadání bylo nutné klást důraz na preciznost vývoje a nezahlcovat tak výkon počítače zbytečnými úkony v aplikaci. Dotazovací příkazy byly uloženy do definičního souboru, ze kterého byly při spuštění aplikace vyčteny. Tímto se vyřešil požadavek na změnu projektu či ŘJ světlometu a zachování funkčnosti SW ovladače.

Po spuštění aplikace se automaticky naváže komunikace mezi zařízením HSX a ŘJ. Komunikace je potvrzena startovacími příkazy pro povolení dotazování a převedení ŘJ do módu nastavování parametrů. Posloupnost příkazů pro vyčítání dat ze světlometu byla vytvořena na základě specifikace. Finální

parametrizace jednotky a nastavení světelné stopy modulu světlometu spočívá v reakci na vyčtené parametry v předchozím kroku.

Zpracování přijatých dat a vizualizace událostí lze považovat za nejdůležitější část SW ovladače. Příkazy byly systematicky rozděleny podle typu komunikace, zda se jednalo o cyklickou komunikaci či komunikaci diagnostickou. Dále byly k příkazům přidány odpovědi a časové trvání daného cyklu. Takto posbíraná data byla zformulována a spojena do jednotlivých logických úrovní pro vizualizace operátorovi během procesu testování světlometu (Obrázek 37). Zobrazená data jsou dle označení OK či NO Error považována za platná a kontrolovaný výrobek je přesunut mezi hotové a otestované kusy bez závady.

### 8.2.3 Nastavení parametrů SW ovladače

Nastavení parametrů pro testování funkcí světlometů probíhá v příslušném definičním souboru. Tento soubor je vždy modifikován podle zadané specifikace, je však pro obě výrobní strany LE/RE stejný. Obsahuje příkazy odesílané do zařízení HSX, kde dochází k jejich správné formulaci, a dále pak do jednotky světlometu, která na daný příkaz odpoví. Většinou se jedná o dotazovací příkazy, při kterých očekáváme v odpovědi právě nastavenou variantu trhu, sériové číslo světlometu či jednotlivých modulů, sériové číslo jednotky, verzi nahraného firmware nebo výčet paměti závad v jednotce. Existují příkazy, které mají za úkol uvést jednotku do určitého stavu, například pro nastavení speciální funkce, vymazání paměti závad nebo příkaz pro uvedení jednotky do módu pro změnu firmware a příprava na flashování nové verze FW (Obrázek 39). Příkazy jsou ze souboru přečteny při spuštění SW Frameworku a jsou uloženy do mezipaměti PC, ze které jsou vyčítány během spuštěné fáze testování.

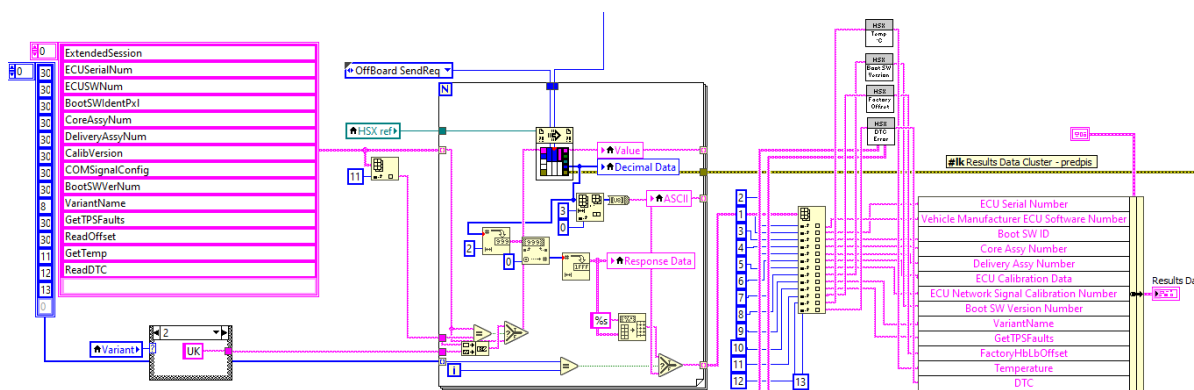
Nastavení samotného SW ovladače probíhá v souboru, avšak s příponou .ini. V tomto souboru se nachází několik sekcí: settings, DTC\_Error a HSX\_Control:

- Sekce settings obsahuje odkaz na .ini soubory dodávané zadavatelem pro přesně daný projekt. Tyto soubory obsahují podrobné nastavení adres, nastavení komunikačních médií a transformaci příkazů z hexadecimální podoby do podoby textového řetězce, které jsou používány v aplikaci.
- Sekce DTC\_Error má za úkol nekontrolovat některé chybné stavy, například pokud není kontrolován celý světlomet, ale pouze jeho část (modul) bude jednotka světlometu hlásit chybu o neúplnosti zapojení. Tento stav je tedy při kontrole funkcionality modulu ignorován.
- Sekce HSX\_Control je nastavení napájení zařízení HSX při řetězovém testování, HSX může být napájeno stále nebo řízeně (PLC).

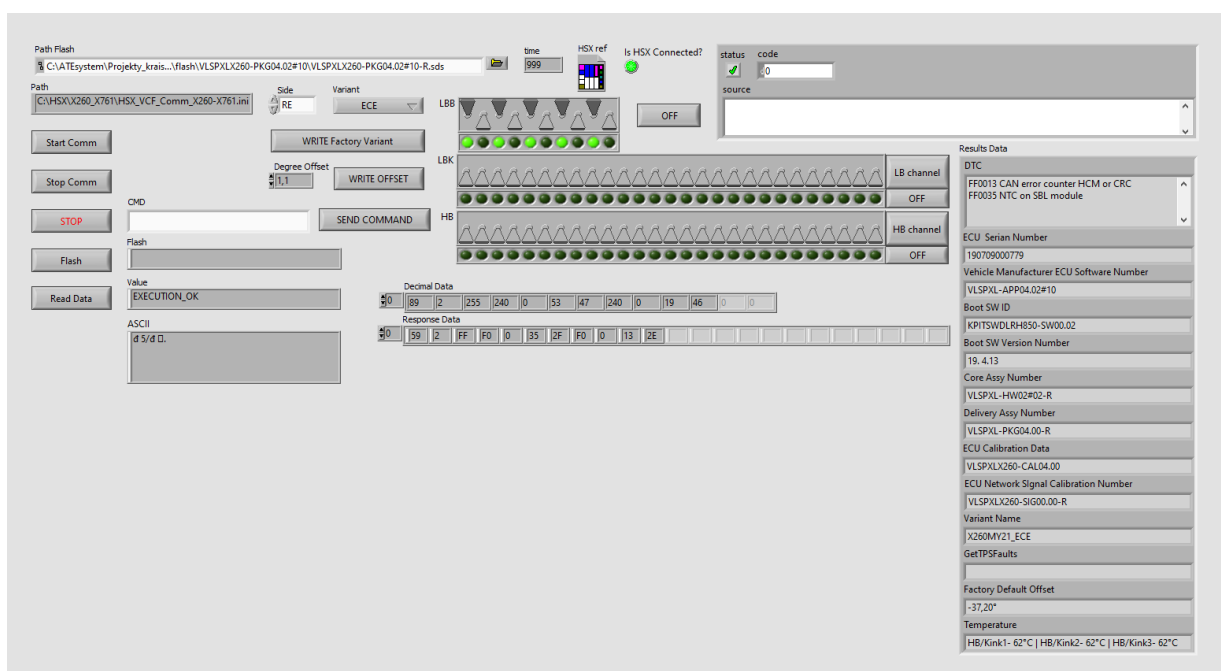
### 8.2.4 Testovací aplikace pro HSX

Testovací aplikace byla vyvinuta pro studii proveditelnosti a testování světlometů i bez příslušných souborů dodaných výrobcem světlometu (Obrázek 39). V testovací aplikaci pro HSX lze jednoduše otestovat příslušné světelné funkce a vyčíst pomocí příkazů informace o jednotce a parametrech světlometu či modulu. Před použitím je třeba modifikovat příkazy, které jsou specifické pro projekt (Obrázek 38) a přidat cestu k souboru pro změnu firmware verze jednotky. Dále stačí správné zapojení konektoru kabeláže ke světlometu, tedy zapojení příslušných vodičů na piny definované v elektrické specifikaci světlometu. Dále je potřeba připojení napájení jednotky a připojení proudových zdrojů na příslušné piny pro oživení světelných funkcí modulu. Aplikaci lze velmi jednoduše ovládat pomocí čelního panelu a nahlížet tak do dat, která jsou přenášena po komunikačním médiu. Testovací aplikace

byla použita jako vzor pro tvorbu několika aplikací při nasazování SW Frameworku do zařízení kontrolujících světelnou stopu, nastavujících parametry ŘJ a definujících verzi FW jednotky světlometu.



Obrázek 38 Algoritmus čtení příkazů v blokovém diagramu testovací aplikace HSX



Obrázek 39 Testovací panel pro komunikaci s ŘJ světlometu

### 8.3 Tvorba SW ovladačů pro zařízení kontroly světelné stopy

Integrace zařízení pro komunikaci s testovací aplikací je podstatnou částí pro chod SW Frameworku a možností získávat hodnoty o jiných datových typech než TRUE a FALSE, jak jsme zvyklí u modulů I/O. Proto se zařízení obsahující procesor, který je schopen vytvořit komunikaci, stále více využívají při průmyslové výrobě a připojují se k nadřazenému řídicímu systému.

Zařízení popisována v této kapitole byla integrována do projektu, při kterém bylo vyráběno zařízení kontroly světelné stopy (Obrázek 40). Integrace zařízení probíhala podle daných předpisů a veškerá funkcionality byla předem otestována.



Obrázek 40 Stanice pro kontrolu světelné stopy

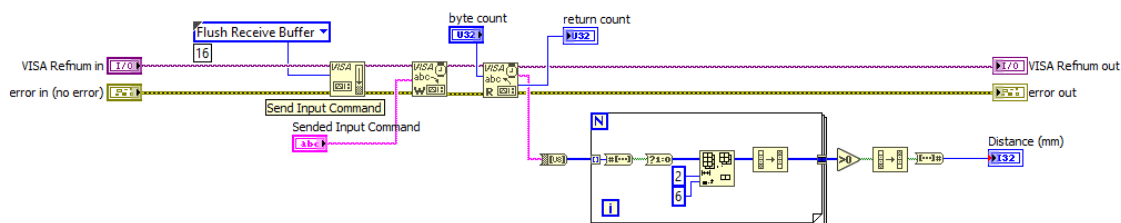
### 8.3.1 SW ovladač dálkoměru

SW ovladač slouží pro integraci dálkoměru do SW framework firemní aplikace. Dálkoměr obsahuje vlastní jednotku a paměť pro uchování stávajícího nastavení a zobrazení měřené hodnoty na displej. Pomocí komunikačního rozhraní RS-485 probíhá komunikace po sériové lince, která umožňuje přenos dat, tedy naměřené hodnoty vzdálenosti ukryté v datovém rámci. Zařízení tohoto typu bylo vytipováno na základě přesného měření vzdálenosti projekční plochy od zrcadla přenášející světelnou stopu testovaného modulu (zdroje světelného paprsku).

Díky možnosti komunikovat se zařízením po vlastní sběrnici lze několik zařízení nezávisle rozdělit a přistupovat k nim v různých časech. Tímto způsobem nejsou komunikace ovlivňovány a aplikace přistupuje ke každému zařízení zvlášť.

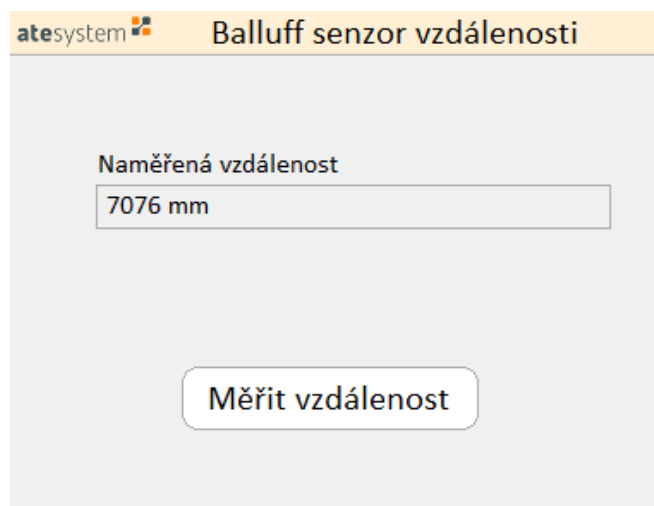
Princip komunikace s dálkoměrem spočívá ve správném nastavení samotného zařízení, kdy zařízení přechází do módu „RS-485“ a je schopno očekávat vstupní zprávu či dotaz na aktuální vzdálenost. Nejprve dojde k inicializaci zařízení, tedy nastavení parametrů komunikace a výčet hodnot těchto parametrů ze souboru nastavení SW ovladače dálkoměru. Vstupní zpráva je tedy pomocí SW ovladače odesílána do připojeného zařízení o přesně dané definici dat. Zařízení na tuto zprávu zareaguje a pokud je dobře nastaveno odpoví zprávou, která je přijímána jako pole o velikosti 3 bajty (Obrázek 41) v kombinaci 1 a 0. Takto přijaté pole na první pohled neobsahuje žádná použitelná data, jelikož ho lze považovat za šifrované. Toto pole je transponované tedy zrcadlově otočené a vždy první dva bity jednoho bytu jsou pouze informativního charakteru o pozici dat. Tyto bity jsou odstraněny a na jejich

pozici jsou posunuty bity následující. Data jsou spojena do jednoho zřetěženého 1D-pole a převedena na hexadecimální hodnotu.



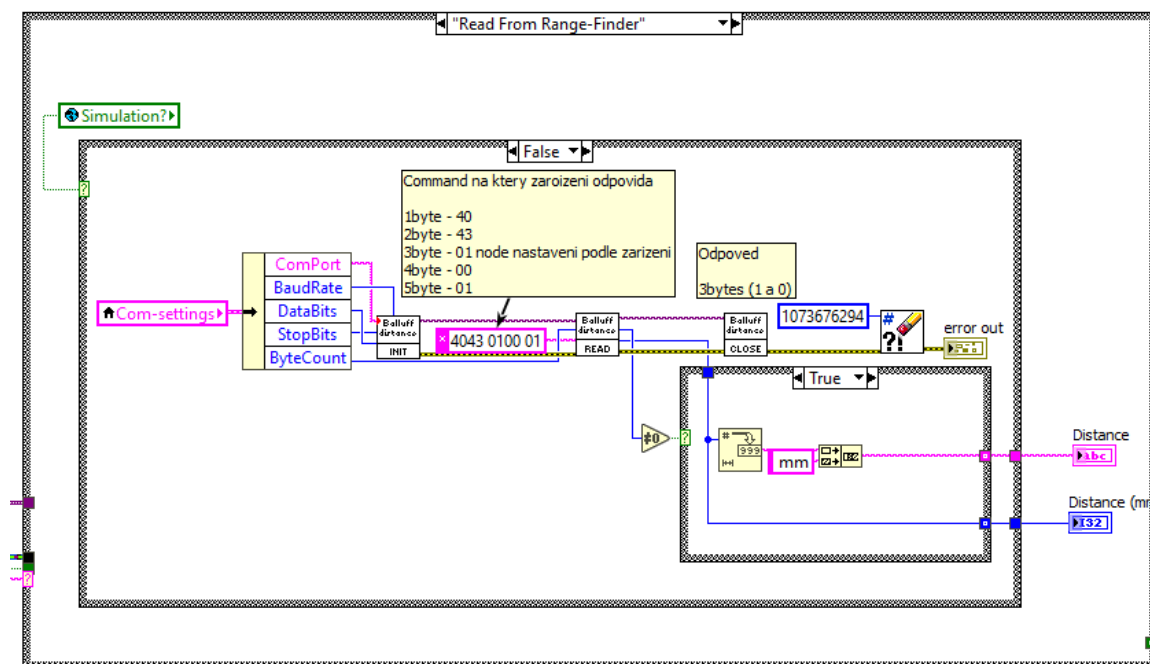
Obrázek 41 Část kódu pro zpracování přijímaných dat ve formě 3bytového pole

SW ovladači náleží i příslušný čelní panel obsahující tlačítko pro okamžité měření vzdálenosti a indikátor naměřené hodnoty. Po stisku tlačítka dojde k odeslání vstupního řetězce definovaného zařízením a následné přečtení přijaté zprávy obsahující změřenou vzdálenost (Obrázek 42).



Obrázek 42 GUI panel SW ovladače

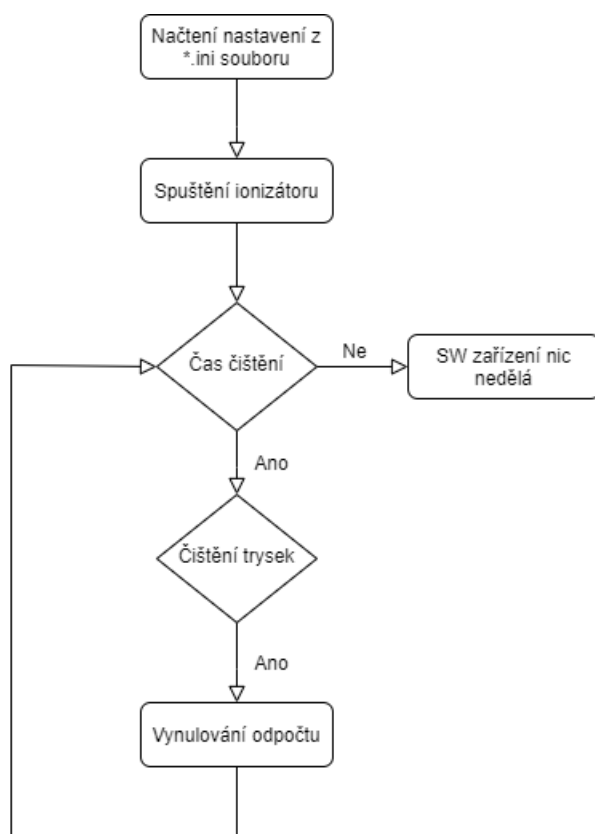
SW ovladač byl implementován podle firemního standardu pro jednotnou integraci vývoje SW komponent pro SW Framework. Při spuštění aplikace dochází k přečtení informací o sériové komunikaci definovaných ze souboru nastavení s příponou .ini. Poté zařízení čeká na stisk tlačítka nebo na zprávu od samotného jádra SW Frameworku či signálu z PLC udávající požadavek na změření vzdálenosti. Po obdržení dotazu z jádra aplikace, je SW ovladačem dálkoměru inicializována komunikace a odeslán řetězec vstupních dat do zařízení. SW ovladač následně očekává odpověď ve formě binárních dat (Obrázek 43). Pokud odpověď do požadovaného času nepřijde, SW ovladač uzavírá komunikaci se zařízením a odesílá hodnotu do zařízení obecných vstupů a výstupů, kde dochází k předání této hodnoty do jádra aplikace. Zařízení přechází zpět do stavu čekání na popud spuštění dalšího měření.



### 8.3.2 SW ovladač ionizátoru

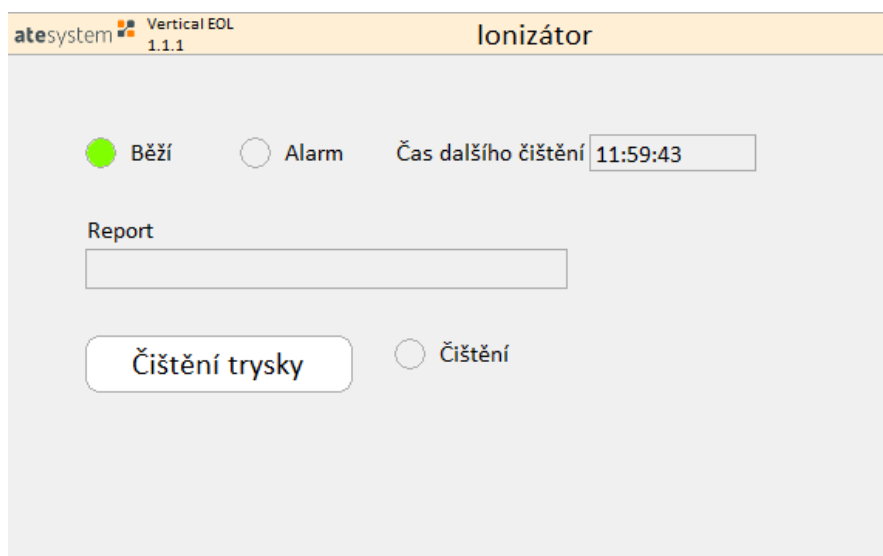
SW ovladač pro ovládání běhu ionizátoru, jehož úkolem je udržovat plochu zrcadla stále lesklou, byl vytvořen pro SW automatizaci ovládání zařízení bez manuálního zásahu. Princip funkcionality zařízení je popsán v UML diagramu (Obrázek 44). Přístup k signálům zařízení je realizován pomocí WAGO Terminálu fungujícím na principu komunikace Modbus plnící funkci I/O modulu celého zařízení. Jelikož má zařízení specifickou logiku a úroveň napětí pro spínání, bylo potřeba přenášet jednotlivé logické úrovně signálu pomocí relé. Pomocí relé se implementuje signál pro zápis i pro čtení do terminálu WAGO, který indikuje aktuální úroveň napětí a jeho SW ovladač převádí tuto úroveň do binární podoby TRUE a FALSE. Na straně zařízení je konektor I/O signálů realizován pomocí konektoru RJ-45, na kterém kombinace definovaných pinů odpovídá logice příslušného signálu.

Zařízení pracuje v několika stavech, které je potřeba identifikovat pomocí SW ovladače a uskutečnit tak řízení chodu zařízení z aplikace. Hlavní stav zařízení je proces spuštění aplikace, načtení všech výchozích nastavení a čekání aplikace na spuštění měření. Pomocí vstupního signálu I/O terminálu je periodicky vyčítán stav zařízení a přenášen do SW ovladače jako hodnota TRUE indikována pomocí prvku čelního panelu. Druhým vstupním signálem je signál alarmu zařízení, při kterém dojde k nečekané události ohrožující běh zařízení. Výstupní signály pro ovládání zařízení je signál kontroly zařízení, což znamená, že je zařízení řízeno pomocí vzdálené periferie (PC) a signál indikace spuštěných trysek pro čištění ionizátoru. Tento signál je ovládán pomocí časového odpočtu čištění, příchozího signálu alarmu, pokud k němu došlo, nebo ke spuštění čištění trysek ze samotné aplikace.



Obrázek 44 UML diagram popsaného zařízení

Čelní panel SW komponenty slouží pro indikaci stavů zařízení a kontroly spuštění čištění trysek ionizátoru (Obrázek 45). Obsahuje časovou značku nesoucí informaci ohledně dalšího čištění trysek. Hodnota času je odpočítávána od spuštění aplikace do dalšího periodického čištění. Indikátory Běží a Alarm jsou reprezentací samotných signálů z ionizátoru. Indikátor čištění reprezentuje probíhající čištění trysek a stav, kdy ionizátor neopracuje v obvyklém režimu, ale v režimu servis. Indikátor Report slouží pro textovou informaci o aktuálním stavu zařízení, například „Probíhá čištění trysek“.



Obrázek 45 GUI SW ovladače ionizátoru

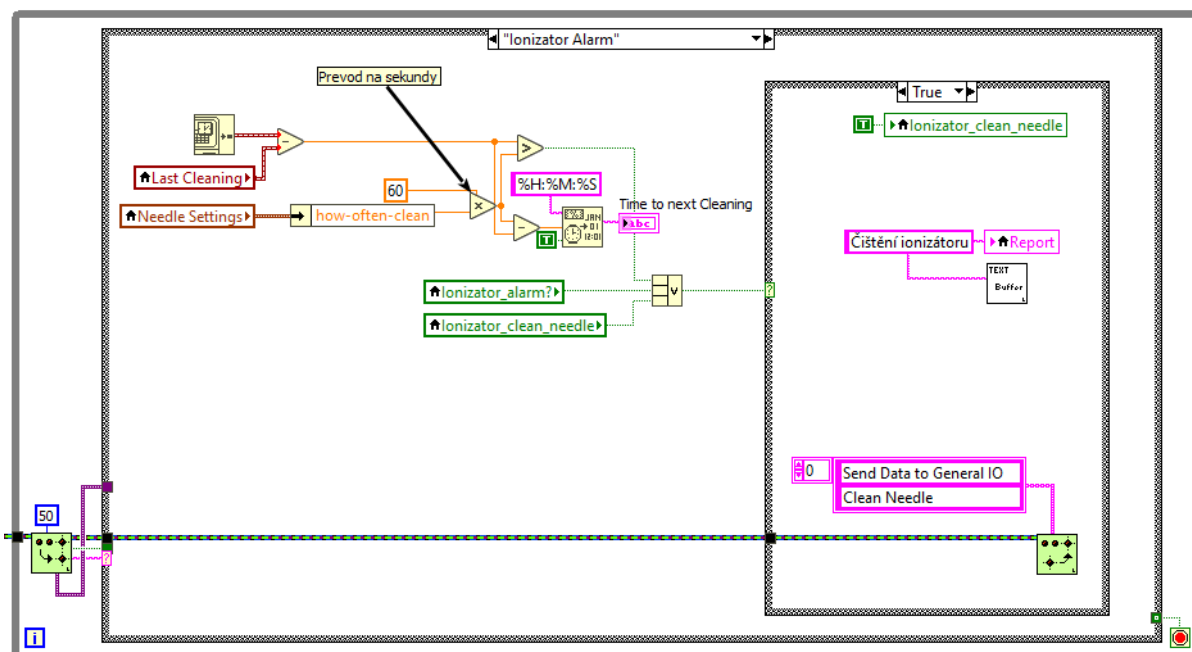
Při implementaci kódu bylo vycházeno z návrhu jednoduchého SW ovladače pro ovládání zařízení, které je nedostupné. Z toho návrhu vyplývá, že funkčnost SW ovladače je kriticky hlídána a bez možnosti ovládání ionizátoru dochází k poškození zrcadlové plochy. Po spuštění aplikace dochází k přečtení názvu signálu ze souboru pro nastavení s příponou .ini. Jedná se o definici signálů pro čtení a zápis informace, která je přenášena v aplikaci právě pomocí definovaných názvů. Názvosloví zvolených signálů musí být přesně definováno a unikátní v celé testovací aplikaci. Ze souboru je nastavován parametr čištění – čas periodicky opakujícího se čištění trysek zadávaný v minutách. Nastavení délky pulzu, tedy podržení hodnoty po definovanou dobu tak, aby byl schopen ionizátor zaznamenat tuto změnu a spustit požadavek na čištění trysek.

Část kódu, při kterém dochází k periodickému vyčítání hodnot signálů, je volána periodicky každých 50 ms. Důvodem je náhlá změna hodnoty signálu a její okamžitá reakce. Signál je složen z několika parametrů: název signálu, označení vstupu a výstupu a jeho číselné označení v souboru všech signálů. Dále je volána část „Ionizator Alarm“, která je popsána níže.

### Ovládání čištění trysek zařízení

Část „Ionizator alarm“ zpracovává informace o příchozích hodnotách signálů či zásahu změny hodnoty tlačítka pro čištění trysek. Pokud dojde k vypršení časové mezery mezi jednotlivými procesy čištění, stisku tlačítka nebo příchozímu alarmu ze zařízení, je provedena změna nastavení řídicího signálu na hodnotu TRUE a tato hodnota je odeslána přes I/O modul do zařízení ionizátoru, které provede již zmiňované čištění. Odpočet je vždy spuštěn ihned po spuštění aplikace (Obrázek 46), která ve stroji běží nepřetržitě.



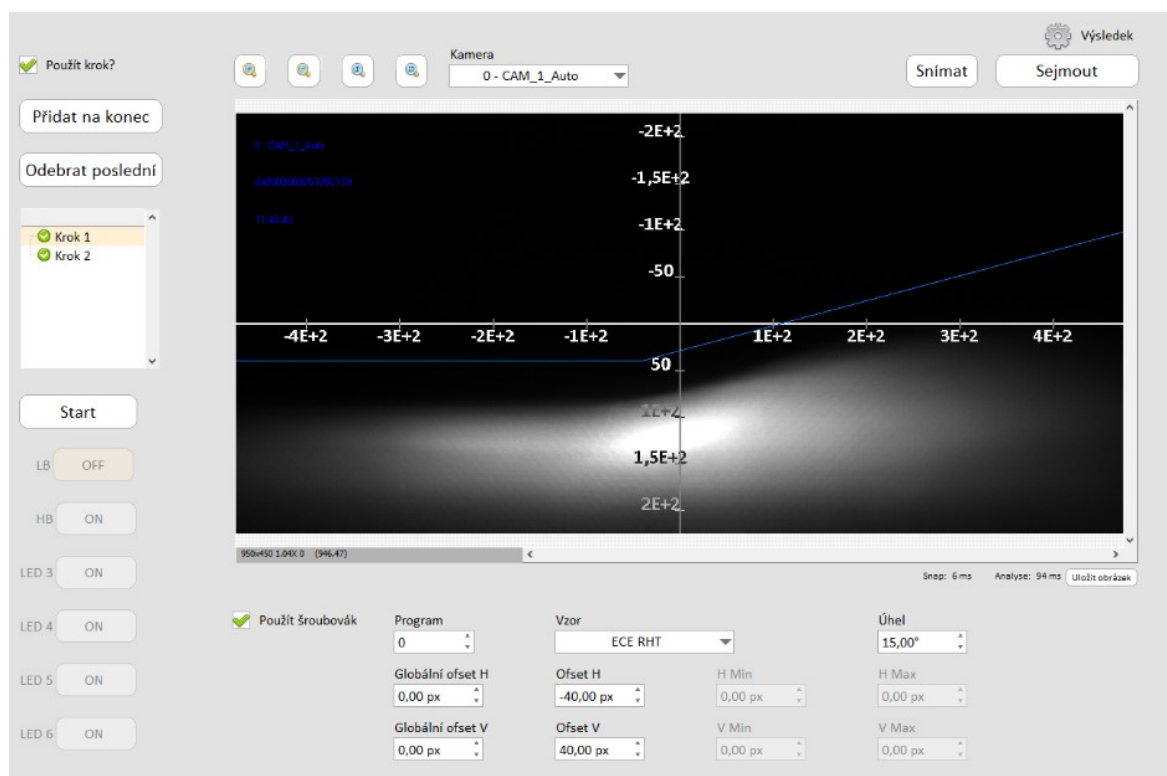


Obrázek 46 Logické zpracování řízného čištění

## 8.4 Algoritmus pro vykreslování křivek, textu

Vývoj SW modulů pro zpracování obrazu a následná implantace zobrazení zajímavých oblastí je nedílnou součástí tvorby kamerových náhledů v aplikacích strojového vidění. Vykreslování obrázců, výpis textu nesoucí informace o měření je důležitým aspektem pro vizualizaci informací na panel SW modulu. SW modul zabývající se pořizováním obrazu ze zvolené kamery uplatňuje příslušné ovladače potřebné pro přenos dat z kamery do PC. Popisovaný panel umožňuje složit sekvenci pro jednotlivé optické testy světelného paprsku a provést jejich vyhodnocení přímo do obrázku panelu. Kroky jsou přidávány pomocí parametrů zobrazených na panelu GUI. Tyto parametry jsou závislé na vybraném zobrazovaném vzoru.

Panel SW modulu Sekvencér kamery umožňuje nastavení parametrů pro zobrazení tzv. „overlays“. Využití vizualizace překryvů je podstatné pro zdůraznění ROI a následnou analýzu obrazu pomocí funkcí obsažených v knihovně pro analýzu a zpracování obrazu „Image Processing“. Zobrazený kříž reprezentující osy lze libovolně posouvat, avšak jeho princip spočívá v definování středu světelné stopy při správném nastavení a umístění kamery (Obrázek 47).

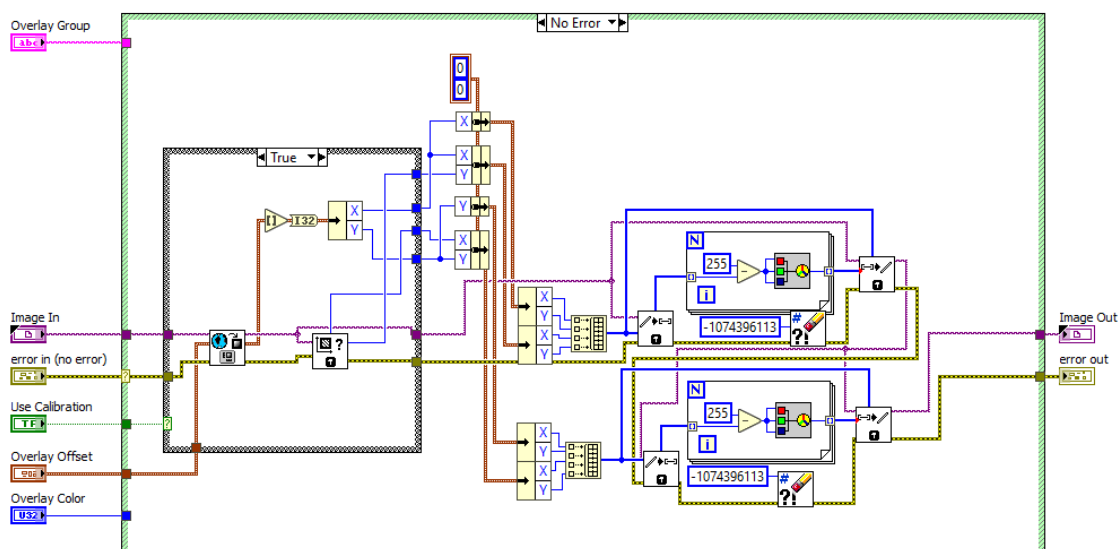


Obrázek 47 Panel pro sestavení sekvence optických testů

### Funkce pro vykreslování překryvů v obraze

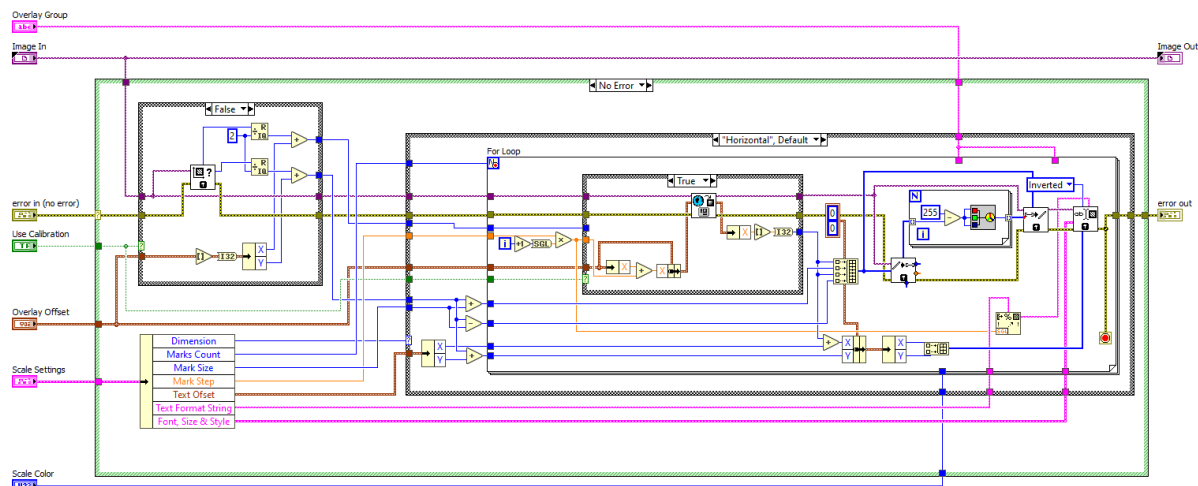
Při práci s již sestavenými funkcemi ve vývojovém prostředí LabVIEW je zapotřebí dobrá znalost těchto funkcí a schopnost implementovat funkce do správné části programu, aniž by docházelo ke ztrátě dat či jejich nekorektní úpravě.

Nejprve je zjištěna velikost vykresleného obrázku v pixelech a poté probíhá již zmiňovaný dopočet umístění obrazce včetně posunutí a barevné interpretace. Jelikož je pořizovaný snímek převeden do černobílé podoby, je i barevná interpretace jednotlivých pixelů vykreslovaného kříže automaticky převedena na hodnotu 0 až 255, jakožto odstíny šedé. Toto dopočítání bylo zvoleno na základě kvalitního překreslování pixelů různé barevné hodnoty. Takto překreslenou hodnotu pixelu dosadíme na původní pozici zvoleného pixelu, avšak s novou barevnou interpretací (Obrázek 48).



Obrázek 48 Vykreslování os do obrazu s převodem barevných hodnot pixelů

Vykreslování jednotek na osy v obraze (Obrázek 49). Nejprve je zjištěna velikost pořízeného obrazu v pixelech, poté je algoritmus převeden k matematickým výpočtům hodnot potřebných pro pole hodnot. Pole hodnot obsahuje pozici, místo, kam má být umístěn text. Pole je zpracováno postupně avšak změna popisu os je vizualizována na konci vykonání této části. Parametry: font, obsah textu, krok jsou nastavovány souborem nastavení a přenášeny pomocí lokální proměnné do části kódu. Data jsou uložena v lokální proměnné a přenesena pomocí konektoru do vnitřní struktury matematické funkce pro vykreslování. Barva vepisovaných popisků os je opět dopočítávána pomocí algoritmu překreslení barevné hodnoty na hodnotu 0 až 255, poté je uložena na původní pozici textu.



Obrázek 49 Vykreslování jednotek na osy

## 9 Nasazení a odladění vyvinutých komponent v průmyslové praxi

V předchozích kapitolách byl popsán můj praktický přínos pro nasazení aplikací v průmyslové výrobě. V této kapitole je popsáno nasazení vyvinutých SW komponent a jejich následné ladění v průběhu výroby. Po implementaci SW komponent pro SW Framework, které byly dle možností otestovány již ve fázi vývoje, bylo náplní pro úspěšné oživení SW a nasazení aplikace do průmyslu příprava průmyslového PC. Vytipování průmyslového PC bylo prováděno dle potřeb připojených zařízení a funkcionality zařízení podle specifikace zákazníka. Většinou při výběru komponent pro sestavení průmyslového PC hraje velkou roli rozpočet vyčleněný pro sestavu PC. Závislost počtu kamer a typ komunikace připojených zařízení upřesňují přidávané komponenty do základní sestavy PC. V dnešní době je hlavním aspektem pro vytvoření testovací aplikace čas optické kontroly jednoho kusu výrobku, který se spolu se zdokonalováním HW neustále zkracuje. Proto se čím dál více přibližujeme použití výkonných grafických karet pro provádění matematických výpočtů během náročného testovacího procesu.

### 9.1 Parametrizace PC, instalace SW

Průmyslové PC pro aplikaci SW Frameworku kontrolující světelnou stopu modulu bylo individuálně sestaveno dle požadavků a účelů dané aplikace. Jako operační systém PC byl zvolen Windows 10 Pro. Počítač obsahoval přidané PCIe karty pro připojení kamer a karty obsahující sériové porty pro připojení zařízení komunikující pomocí rozhraní RS-232 či RS-485. Bylo provedeno prvotní spuštění a konfigurace OS Windows. Následovalo zálohování čisté instalace OS bez instalovaných SW. Instalace SW potřebného pro funkci testovací aplikace byla zadokumentována a odzkoušena. SW potřebný pro testovací aplikaci a komunikaci zařízení obsahoval: SW od společnosti NI vývojové prostředí LabVIEW, Vision Development Module a Vision Acquisition Software a ovladače průmyslových kamer společnosti Basler. SW pro konfiguraci a správu záložního zdroje, SW pro komunikaci připojených zařízení například konfigurační SW pro WAGO Terminál a SW pro čtečky kódů, karet atd.

### 9.2 Testování

Testování vyvinutých SW komponent probíhalo již během vývoje. Firemním standardem je vytvořit simulační část každého SW ovladače a otestovat tak chování vyvinutých SW komponent. Součástí je testování vytvořené aplikace v simulačních podmínkách pomocí testovacích snímků pořízených při studii proveditelnosti.

Následně byla všechna zařízení připojena a postupně otestována, odladěna na takto vytvořených SW ovladačích. Každá takto vytvořená komponenta obsahovala i uživatelské rozhraní, jelikož vizualizovala uživateli měřené či přečtené parametry. Díky možnosti otestovat SW ovladače v simulaci mohly být SW komponenty po připojení zařízení a nastavení komunikace ihned zprovozněny a nasazeny.

Před nasazením aplikací do provozu bylo procesem testování i spouštění sekvence testů a vyhodnocování na testovacích kusech dodaných zákazníkem. Pokud však nebyly testovací kusy dodány v průběhu vývoje, došlo k validaci správného chování aplikace během testovací výroby u zákazníka. Na testovacích kusech bylo provedeno odladění SW Frameworku a malé úpravy požadované zákazníkem pro lepší informativnost testované sekce.

Při aplikaci, kdy byla vyvinuta SW komponenta hranového detektoru, muselo být otestováno správné rozhodování o Ok a NOK kusech. Po konzultaci se zákazníkem došlo k úpravě limitu úspěšnosti a změně některých parametrů pro nastavení detekce hrany, k výraznému zlepšení úspěšných výsledků test

a zvýšení počtu kontrolovaných kusů označovaných značkou OK (SW modul kontroly nanesené pasty na skle automobilu).

Při aplikaci kontroly světelné stopy, bylo prováděno testování u zákazníka za přítomnosti několika světelných modulů. Díky tomuto testování bylo nově implementováno dialogové okno informující operátora o NOK kusech. Upravená aplikace byla nasazena do průmyslové výroby a nyní probíhá fáze optické kontroly výrobku na testovacích kusech (Tvorba SW ovladačů pro zařízení kontroly světelné stopy).

Zpětná vazba ze strany zákazníka je v obou případech kladná a uvažuje se o další spolupráci nasazení SW Frameworku na zařízení o podobném principu detekce či optické kontroly výrobku.

## 10 Ověření funkce vyvinutého SW a zhodnocení

Ověření funkce všech vytvořených aplikací nebo jejich částí bylo nedílnou součástí diplomové práce a probíhalo zároveň při testování. Během tohoto procesu byly odladěny veškeré nedostatky způsobené změnami specifikace. Postupnými změnami v zapojení či vylepšením funkcionality nebo zobecněním kódu pro následné použití v jiných aplikacích došlo ke zlepšení stability aplikace při kritických podmínkách provozu. Ověření probíhalo vždy zadavatelem či vedoucím projektu, který zadaný úkol zkontroloval. Předvedení funkčnosti bylo vždy provedeno a řádně zdokumentováno. SW komponenty vyvinuté během diplomové práce jsou nyní nasazeny v průmyslové výrobě. Reakce na nasazení SW komponent či aplikací strojového vidění měly kladné ohlasy ze stran zákazníka a lze je hodnotit za ověřené a uzavřené. Ověření funkce vyvinutých SW ovladačů probíhalo ihned po implementaci. Tyto ovladače plní funkci řízení fyzického zařízení přes komunikační médium a zjednodušení obsluhy zařízení. Obsluha tedy řídí zařízení z jednoho místa pomocí testovací aplikace. SW komponenty obsluhující princip určitého testu v aplikaci byly většinou testovány během nasazení aplikace u zákazníka za přítomnosti testovacích kusů pro vytřídění OK a NOK kusů, tedy parametrizační komponenty pro správné rozřazování výrobků dle specifických požadavků.

Hodnocení vyvinutých komponent je tedy kladné. Díky nasazení v průmyslové výrobě a zpětné vazby zákazníka lze soudit, že aplikace splňují svou funkci a pracují dle předem stanovených požadavků. SW komponenty, které byly během diplomové práce vyvinuty jsou zdokumentovány pro možnost použití i v jiných aplikacích strojového vidění. SW framework testovací aplikace je nasazen v průmyslové výrobě a plní funkci optické kontroly průmyslového výrobku při jeho fázi výroby. Aplikace jsou nasazeny na zařízeních pracujících v nepřetržitém výrobním procesu, a ze strany firmy je možná vzdálená podpora či pomoc při jakýchkoliv dotazech či změnách výroby.

## Závěr

Výsledkem diplomové práce jsou vyvinuté SW komponenty nasazené v průmyslové výrobě, které plní svou funkci, na SW Frameworku. Diplomová práce obsahuje teoretický rozbor kamerového řetězce pro optickou detekci výrobku, možnosti výběru hardwaru pro snímání obrazu podle různých parametrů, návrh a implementace softwaru pro řízení, zpracování obrazu, ovládání zařízení, která jsou potřebná pro čtení dat z výrobku nebo měření hodnot pro správný průběh testování a dodržení konstantních podmínek pro opakování testování. Diplomová práce popisuje vývoj SW ovladačů pro tyto zařízení včetně návrhu uživatelského rozhraní a statistického zpracování dat ve vývojovém prostředí LabVIEW. Součástí práce je testování a odlaďování vyvinutých částí na průmyslové výrobní lince zákazníka.

Výběr hardwarových komponent pro snímání obrazu byl volen v souladu s požadavky automatického testeru (poloha a velikost výrobku, typy hledaných artefaktů a světelné podmínky). Součástí výběru bylo testování vytípaných komponent na vzorcích dodaných zákazníkem, provádění studie proveditelnosti a odhadnutí časové náročnosti vývoje SW pro požadované optické testy.

SW komponenty byly vždy vyvinuty v souladu s firemním standardem pro návrh SW ovladačů nebo SW modulů obsahující příslušné GUI v panelu firemní aplikace a souboru pro nastavení parametrů funkčnosti SW komponenty. Náplní tvorby SW ovladačů byla integrace zařízení s určitým typem komunikace, tedy o otestování komunikace a typu posílaných dat (rozklíčování). Dále o vývoj SW ovladače příslušného zařízení pro komunikaci. Veškerý vývoj SW komponent byl spojen s daným typem projektu a všechny vyvinuté části jsou nasazené při výrobě v automatizovaném testeru. Vývoj SW modulů pro optickou kontrolu byl použit ve SW Frameworku pro konkrétní aplikaci testeru. Při tomto vývoji bylo také potřeba modifikovat sekvenci testů a vytvořit tak soubor neboli skript pro logické pořadí a návaznost testů.

K podstatné části patřilo testování a ladění software. Testování probíhalo během vývoje, následně přímo v průmyslové výrobě. Během testování byly odstraněny drobné chyby či poupravena funkcionality SW komponenty pro správné plnění své funkce. Dále také došlo k upravení aplikace testeru po změnách požadavků zákazníka, či přidání okna výpisu informací o naměřených parametrech testů.

Na závěr diplomové práce lze konstatovat, že bylo dokončeno několik projektů realizovaných v průmyslové výrobě. Stále probíhá podpora v oblasti úpravy SW Frameworku nasazeného u zákazníka s možností implementace dalších funkcionalit do stávajících SW komponent. SW komponenty jsou připraveny pro použití v následujících projektech a jejich vývoj je pečlivě zdokumentován.

Pro realizaci diplomové práce jsem využil poznatky získané praxí ve firmě: volba hardware pro kamerovou inspekci, správné nasvícení a zviditelnění kontrolovaných artefaktů výrobku. Mezi znalosti získané během studia patří: návrh a realizace software, vytvoření grafického rozhraní či rozvržení pracovního postupu neboli rozfázování projektu. Diplomová práce mě obohatila o nové zkušenosti spojené s integrací zařízení do průmyslové výroby, tvorbou SW testovacích aplikací pro realizaci optické kontroly průmyslového výrobku, zkušenosti s laděním aplikace a testováním výrobků v průmyslové výrobě.

## Literatura

- [1] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. BEN - technická literatura, 2008. ISBN 978-80-7300-245-9.
- [2] HOTARŤ, Vlastimil. *Úvod do problematiky strojového vidění*. Liberec: Liberec : Technická univerzita v Liberci, 2015, 2015. ISBN 978-80-7494-156-6.
- [3] BRESS, Thomas J. *Effective LabVIEW programming*. Allendale: National Technology & Science Press, c2013. ISBN 978-1-934891-08-7.
- [4] Analýza obrazu – Strojové vidění. *Analýza obrazu* [online]. Copyright © [cit. 24.11.2019]. Dostupné z: <http://www.analyza-obrazu.cz/strojove-videni/>
- [5] POLÁK, Karel. Sběrnice CAN. *Sběrnice CAN* [online]. 16.6.2003 [cit. 2019-11-30]. Dostupné z: Elektrevue [online]. Dostupné z: <http://www.elektrevue.cz/clanky/03021/index.html>
- [6] SUTORÝ, Tomáš. LIN (Local Interconnect Network). *LIN (Local Interconnect Network)* [online]. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2004, 10.3.2004 [cit. 2019-11-30]. Dostupné z: <http://www.elektrevue.cz/clanky/04012/index.html>
- [7] CMOS vs CCD sensors. *Which is better? Comparison of the Image Pixel Sensor Architectures* [online]. [cit. 2020-02-24]. Dostupné z: [https://meroli.web.cern.ch/lecture\\_cmos\\_vs\\_ccd\\_pixel\\_sensor.html](https://meroli.web.cern.ch/lecture_cmos_vs_ccd_pixel_sensor.html)
- [8] POLÁK, Karel. In: *A Practical Guide to Machine Vision Lighting* [online]. NI, 2019 [cit. 2019-11-30]. Dostupné z: <http://www.ni.com/cs-cz/innovations/white-papers/12/a-practical-guide-to-machine-vision-lighting.html>
- [9] PARKER, J. R. *Algorithms for Image Processing and Computer Vision*. Druhé. Velká Británie: John Wiley, 2010. ISBN 9780470643853.
- [10] Machine Vision. *Community NI: Machine Vision* [online]. 2013, 2013 [cit. 2019-11-30]. Dostupné z: <https://forums.ni.com/t5/Machine-Vision/Pointing-out-a-particle-from-Freehand-ROI/td-p/2671503?profile.language=en>
- [11] KWON, Kye-Si a Steven READY. *Practical guide to machine vision software: an introduction with LabVIEW*. Weinheim: Wiley-VCH, 2015, 300 s. ISBN 978-3-527-33756-9.
- [12] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image processing, analysis, and machine vision*. 4th edition. United States of America: Cengage Learning, 2015, 872 s. International Edition. ISBN 978-1-133-59369-0.
- [13] SIMARD, Jean-Marc. *Exaprom PDF 2.0: PDF report generation toolkit for LabVIEW* [online]. In: . 27. září 2010 [cit. 2020-03-09].
- [14] *CoaxPress* [online]. JIIA, 2020 [cit. 2020-04-17]. Dostupné z: <http://www.coaxpress.com/>
- [15] Edmund Optics: Introduction to Modulation Transfer Function. *Introduction to Modulation Transfer Function* [online]. 101 East Gloucester Pike, Barrington, NJ 08007-1380 USA: Edmund, 2020 [cit. 2020-04-17]. Dostupné z: <https://www.edmundoptics.com/knowledge-center/application-notes/optics/introduction-to-modulation-transfer-function/>
- [16] K ČEMU JE DOBRÁ VINĚTACE A PROČ S NÍ PRACOVAT?: Typy a triky. *K ČEMU JE DOBRÁ VINĚTACE A PROČ S NÍ PRACOVAT?* [online]. mujolympus, 2017, 8. srpna 2017 [cit. 2020-04-17]. Dostupné z: <https://www.mujolympus.cz/blog/tipy-a-triky/3284/k-cemu-je-dobra-vinetace-a-proc-s-ni-pracovat>
- [17] Slovník pojmů. *Megapixel* [online]. [cit. 2020-04-18]. Dostupné z: <https://www.megapixel.cz/svetlnost-objektivu>



- [18] GŘEŠ, Tomáš. *KOMUNIKAČNÍ ROZHRANÍ VE STROJOVÉM VIDĚNÍ* [online]. [cit. 2020-04-18]. Dostupné z: <http://kamery.atesystem.cz/know-how/srovnani-nejpouzivanejsich-komunikacnich-rozhrani-ve-strojovem-videni/>
- [19] *Barevné a Infračervené filtry* [online]. [cit. 2020-04-18]. Dostupné z: <http://kamery.atesystem.cz/produkty/podle-vyrobce/schneider-kreuznach/filtry/barevne-a-nir-filtry/>
- [20] VisualApplets – Visual FPGA Programming of Image Processing Applications. *Basler* [online]. [cit. 2020-04-19]. Dostupné z: <https://www.baslerweb.com/en/products/frame-grabber-portfolio/visualapplets/>
- [21] BILÍK, Petr a kolektiv, 2012. *Měřicí systémy a jejich programování: učební text a návody do cvičení*. Ostrava

## Přílohy

### Elektronická příloha

Adresář elektronické přílohy obsahuje podadresáře:

- *Barevne\_podbarveni\_pixelu* – obsahuje testovací funkce pro překreslení barevnosti pixelu dané pozice a výpočet časové náročnosti převodu
- *Hranovy\_detektor* – obsahuje funkce pro detekci hran v oblastech zájmu
- *HSX* – obsahuje funkce analýzy příkazů z ŘJ světlometu, obsahuje testovací funkce převodu příkazů do jiné číselné soustavy
- *SW\_modul\_sekvencer\_testu\_kamery* – obsahuje funkce pro vykreslování daných artefaktů do obrazu
- *SW\_ovladac\_dalkomeru* – obsahuje funkce pro navázání komunikace se zařízením a výčet vzdálenosti
- *HSX\_Manual\_Softing.pdf* – obsahuje uživatelský manuál pro práci se zařízením HSX interface včetně zapojení konektoru a technických dat
- *PDF\_report\_generation\_toolkit\_for\_LabVIEW\_Manual.pdf* – obsahuje návod pro tvorbu reportů ve formátu \*.pdf pomocí vývojového prostředí LabVIEW